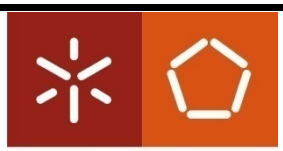
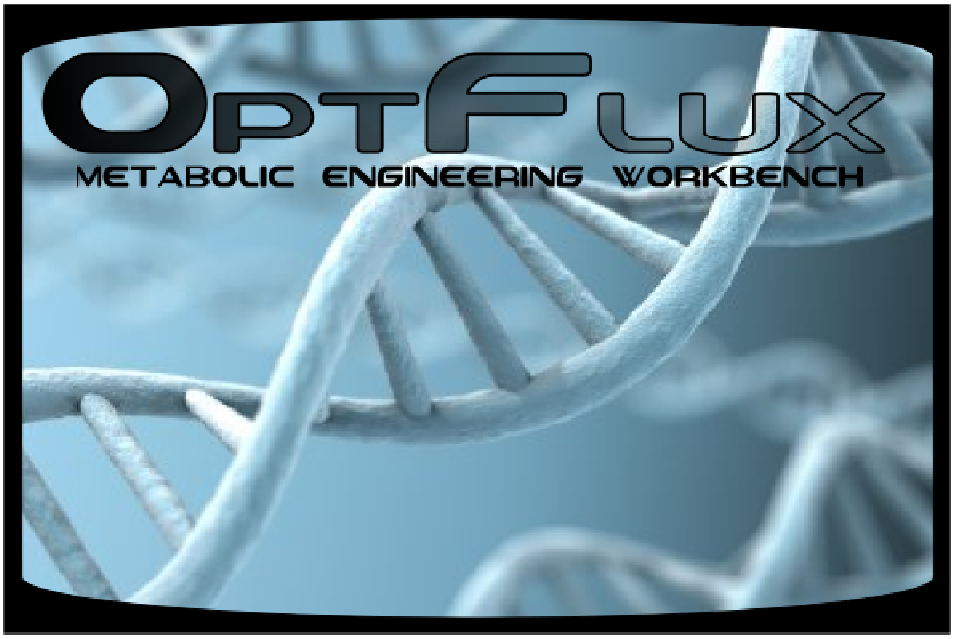


Universidade do Minho

IBB-CEB – INSTITUTE FOR BIOTECHNOLOGY AND BIOENGINEERING – CENTRE OF BIOLOGICAL ENGINEERING
CCTC– COMPUTER SCIENCE AND TECHNOLOGY CENTER
SCHOOL OF ENGINEERING
UNIVERSITY OF MINHO

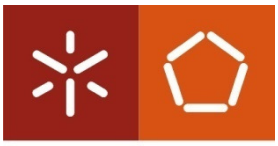
USER MANUAL



Document revision based on OptFlux v1.35

OPTFLUX

METABOLIC ENGINEERING WORKBENCH



Universidade do Minho

IBB-CEB – INSTITUTE FOR BIOTECHNOLOGY AND BIOENGINEERING – CENTRE OF BIOLOGICAL ENGINEERING
CCTC– COMPUTER SCIENCE AND TECHNOLOGY CENTER
SCHOOL OF ENGINEERING
UNIVERSITY OF MINHO

© *IBB-CEB/CCTC*
All rights reserved.

TABLE OF CONTENTS

TABLE OF CONTENTS **3**

LICENSES **6**

1 | INTRODUCTION **7**

2 | BASIC CONCEPTS AND INTERACTION **8**

2.1 – Datatypes and Operations 8
2.2 – Plug-in Based Framework 8
2.3 – User Interaction and the MVC 9

3 | *OPTFLUX* – GETTING STARTED **11**

3.1 – Creating projects and loading data 11
3.1.1 – Using the New Project Wizard 12
3.2 – Model graph visualization 18
3.3 – Simplifying models 19
3.4 – Identifying essential genes 20
3.5 – Using Environmental Conditions 21
3.6 – Simulation 22
3.6.1 – Using the Simulation Wizard 23
3.6.2 – Flux Variability Analysis (FVA) 24
3.7 – Optimization with Evolutionary Algorithms 25
3.7.1 - Solutions Encoding 25
3.8 – Optimization with Simulated Annealing 27
3.9 – Optimization with Local Search 30
3.10 – Using the Optimization Wizard 31

4 | OPERATIONS – COMPLETE REFERENCE **32**

4.1 – File 32
4.1.1 – New Project Wizard 32
4.1.2 – Load 34
4.1.3 – Save 35

4.1.4 – Export 36
4.1.5 – Quit 37
4.2 – Simulation 37
4.2.1 – Simulation Wizard 38
4.2.2 – Wild Type Simulation 39
4.2.3 – Mutant Simulation 39
4.2.4 – Flux Variability Analysis 40
4.2.5 – Environmental Conditions 40
4.3 – Optimization 41
4.3.1 – Optimization Wizard 41
4.3.2 – Preprocessing 43
4.3.3 – EA Optimization 44
4.3.4 – SA Optimization 45
4.3.5 – Local Optimization 45
4.4 – Options 46
4.4.1 – Project 46
4.5 – Help 46
4.5.1 – About 46

5 | DATATYPES – COMPLETE REFERENCE 47
5.1 – *AIBench* types of Datatypes 47
5.2 – *OptFlux* Datatypes 47
5.2.1 – Critical Reactions (CriticalBox) 47
5.2.2 – Original Metabolic Model (DataBox) 48
5.2.3 – Environmental Conditions (EnvironmentalConditionsSet) 48
5.2.4 – Equivalent Variables (EquivalencesBox) 48
5.2.5 – Model Graph (ModelBox) 49
5.2.6 – OptimizationResults 49
5.2.7 – Project 49
5.2.8 – Simplified Metabolic Model (SimplifiedDataBox) 50
5.2.9 – Simulation Results 51
5.2.10 – Variability Analysis Results 51
5.2.11 – Zero Fluxes (ZeroBox) 51

6 | VIEWERS / EDITORS – COMPLETE REFERENCE 52
6.1 – Viewers or Editors 52
6.2 – *OptFlux* Viewers and Editors 53
6.2.1 – Critical Reactions Viewer 53

6.2.2 – Equivalent Variables Viewer 54
6.2.3 – Environmental Conditions Viewer 55
6.2.4 – Metabolites Viewer 56
6.2.5 – Model Graph Viewer / Graph Overlap 57
6.2.6 – Model Properties Viewer 58
6.2.7 – Optimization Solutions Viewer 59
6.2.8 – Reactions Viewer (Flux Bounds) 60
6.2.9 – Reactions Viewer (Stoichiometric Model) 61
6.2.10 – Steady-State Equations Viewer (Stoichiometric Model) 62
6.2.11 – Simulation Solution Viewer 63
6.2.12 – Variability Analysis Viewer 64
6.2.13 – Zero Variables Viewer 65

7 | FILE FORMATS 66
7.1 – Flat Files for models 66
7.1.1 – Flux Bounds File 66
7.1.2 – Sparse Matrix File 66
7.1.3 – Full Matrix File 67
7.1.4 – Metabolites File 67
7.2 – Flat Files for Other Components 68
7.2.1 – Critical Reactions File 68
7.3 – SBML files 68
7.3.1 – Pure SBML Files 68
7.3.2 – CellDesigner SBML Files 68

LICENSES

- **For this user manual:**

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0 or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

- **For the OptFlux software:**

Copyright 2009
IBB-CEB - Institute for Biotechnology and Bioengineering - Centre of Biological Engineering
CCTC - Computer Science and Technology Center
University of Minho

This is free software: you can redistribute it and/or modify
it under the terms of the GNU Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This code is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the GNU Public License for more details.

You should receive a copy of the GNU Public License
along with the code. If not, see http://www.gnu.org/licenses/

Created inside the SysBio Research Group (http://sysbio.di.uminho.pt)

1 | INTRODUCTION

The *OptFlux* application includes a number of tools to support *in silico* metabolic engineering. The application allows the user to load a genome-scale stoichiometric model of a given organism. This will serve as the basis to simulate the wild type and mutants (original strain with a set of selected gene deletions). The simulation of these strains will be conducted using a number of approaches (e.g. Flux-Balance Analysis (FBA), Minimization of Metabolic Adjustment (MoMA) or Regulatory On/Off Minimization (ROOM)) that allow the set of fluxes in the organism's metabolism to be determined, given a set of environmental constraints. The software also includes a number of optimization methods to reach the best set of gene deletions given an objective function, typically related with a given industrial goal.

This application is complemented by an independent visualization tool, named *BioVisualizer*, which allows the visualization of graphs representing metabolic networks. These graphs have a number of distinct node types (e.g. metabolites, enzymes, reactions) and connections. One of the major features of this tool is the ability to associate numerical values to the different types of nodes and edges in the graph.

The integration of the two applications allows the visualization of the metabolic network, superimposed by the values of the fluxes of a given simulation (a feature well integrated in the *OptFlux* application). The values of the fluxes for the wild type and different mutants can, therefore, be adequately visualized in this way, providing useful analysis tools for the researchers.

Additionally, this software is compatible with several SBML (Systems Biology Markup Language) standards, allowing the utilization of models stored in public databases (e. g. like BioModels) or built in other software tools (e. g. like Cell Designer) both for simulation and visualization.

The *OptFlux* application is being developed taking as a basis the *AIBench* framework. This is an environment for the development of Data Mining / Bioinformatics tools, using the Java programming language. The details of this project, a collaboration between the universities of Minho (Portugal) and Vigo (Spain), as well as updated documentation can be found at the web site: http://www.aibench.org.

This document briefly explains the functionalities of the *OptFlux* application and the way it can presently be used. This is however, still a preliminary version of the documentation. Therefore, it should serve as the basis to understand the potential of the approach and be faced as a basis for future development.

2 | BASIC CONCEPTS AND INTERACTION

2.1 – DATATYPES AND OPERATIONS

Every application built based on *AIBench* is organized around the concepts of datatypes and operations, defined as follows:

- **DATATYPES**: define the types of data that are of interest to a given application. For each data object, one or more visualizers can be defined to show its content to the user in a given perspective. Data objects can have a hierarchy, where a given object A contains objects B and C (these objects are called compound objects).

  The set of data objects and their hierarchy in a given application are shown in a tree (the **clipboard**), typically appearing in the left side of the screen. In this tree, compound objects can be opened to show their contents (a list of other data objects). When a given data object is selected (double clicked), the available viewers (if any) are launched in the right area of the screen.

- **OPERATIONS**: each operation defines a function that takes zero or more data objects as its inputs and can create as an output zero or more data objects and/or merely change the input data objects. Operations can be accessed through the menu options, being typically grouped in several menus and sub-menus. Operations can also be run from the clipboard, by right clicking (context menus) a data object (this will show the list of all available operations using that data object as an input).

2.2 – PLUG-IN BASED FRAMEWORK

AIBench uses *Platonos* (http://platonos.sourceforge.net/), a powerful plug-in engine. This gives *AIBench's* built applications such as *OptFlux* the power to be easily upgraded or extended.

- **Apply new plug-ins to *OptFlux*:**

Currently, the insertion of new plug-ins has to be done manually, in a simple process:

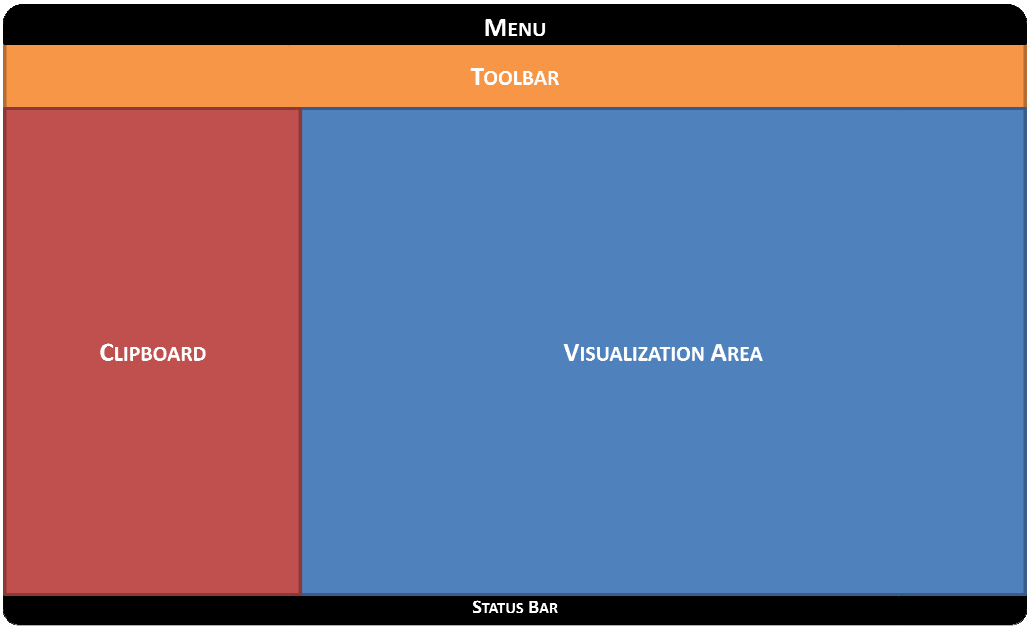
1. - Download the plug-in from the *OptFlux* website (check for versions);
2. - Copy the downloaded archive to $OPTFLUX_HOME/plugins_bin;
3. - Restart OptFlux.

2.3 – User Interaction and the MVC

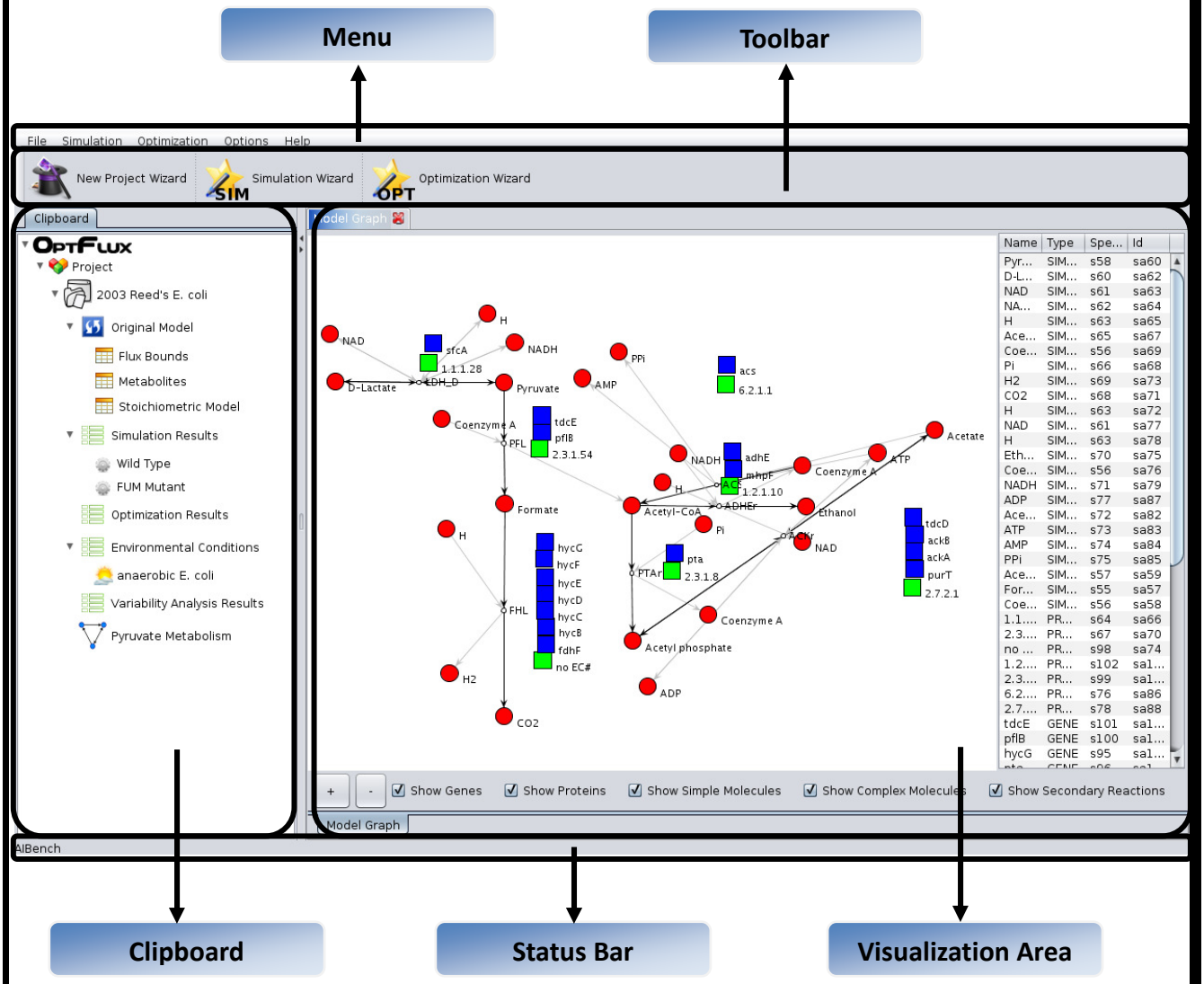
Since *OptFlux* is an *AIBench* based application, the user interaction is thought to be as simple as possible. The *Model-View-Controller* (MVC) architectural pattern has been used in every step of development of *AIBench* as well as of *OptFlux,* resulting in a great deal of decoupling between the operational data and the views.

As mentioned before, in *OptFlux* a View is related to a given Datatype. If there is no View associated with a Datatype, a Default View is launched (a *bean inspector*). The Views will, by default, be launched on the right side of the work area.

The original layout of the components can be observed in the diagram bellow. An area denoted as *other tools* is also available. It is used to launch several different tools, usually not directly related with the problems solved in *OptFlux*. By default that area is turned off, resulting in an expansion of the working area, but it can be easily turned back on.



A screenshot of *OptFlux* is shown bellow to give an idea of the default layout of the system. Looking closely it can be seen that once a data object is selected in the Clipboard, the corresponding Viewer is launched on the right side.

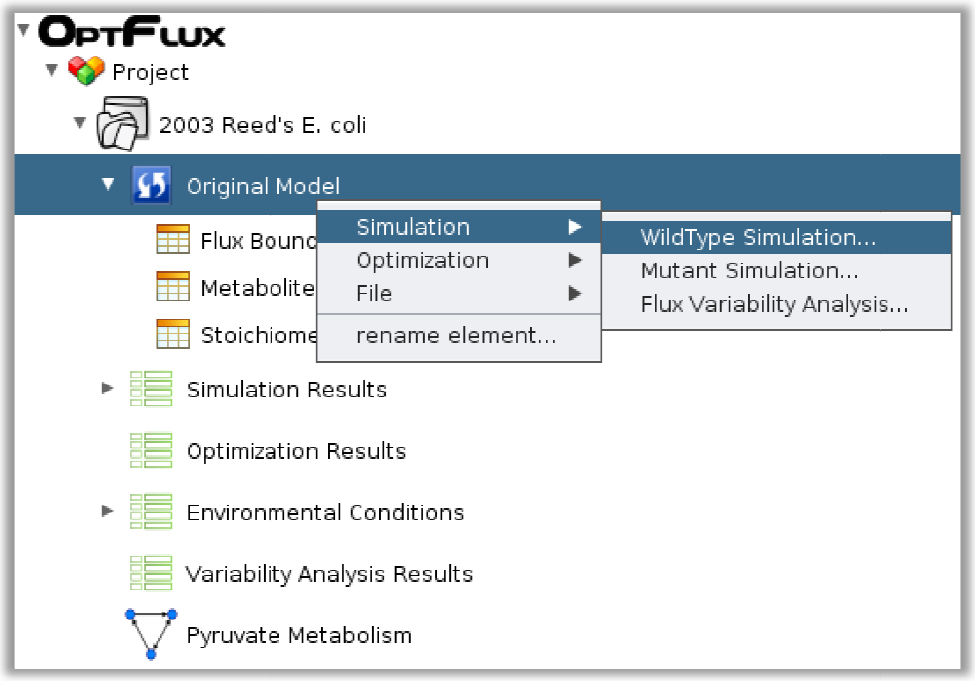


3 | *OptFlux* – Getting Started

This chapter will focus on the main tasks that can be performed with *OptFlux*. The main techniques that *OptFlux* uses will also be briefly explained. A full list of operations, datatypes, viewers and other details will be presented in chapters 4, 5 and 6.

3.1 – Creating projects and loading data

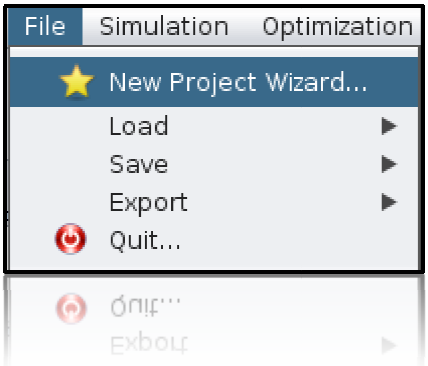
The **Project** is the main data object in *OptFlux*. It is a compound object, used to hold other objects such as a metabolic model, its visualization graph and the results of the pre-processing, simulation and optimization operations conducted over this model. In this version, the full content of a project (in this case the *Escherichia coli* project) is the following:

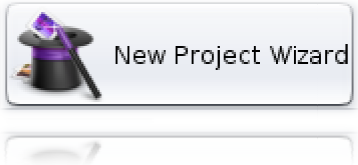


- The original metabolic model object that contains information on the reactions, fluxes and metabolites involved, including the stoichiometric matrix;
- A simplified metabolic model (OPTIONAL) – generated when a simplification of the original model is requested. The information contained is the same as in the original metabolic model, plus information on the equivalent variables and zero values discovered during the simplification procedures.
- A graph representation of the model (loaded from a SBML file) that can be used to visualize the model (or a part of it);
- A list of critical reactions, i.e. reactions that cannot be knocked out to maintain biomass production;
- A list of results from model simulations (using FBA/MoMA/ROOM) of wild type or mutant strains;
- A list of results from optimization algorithms (objects of type "Solution Set"), that hold sets of solutions of the previous type;
- A list of variability analysis results (using Flux Variability Analysis);
- Lists of different environmental conditions that can be used in the simulation of the model

3.1.1 – Using the New Project Wizard

The **New Project Wizard** is accessible both through the *File Menu* and the *Toolbar*.



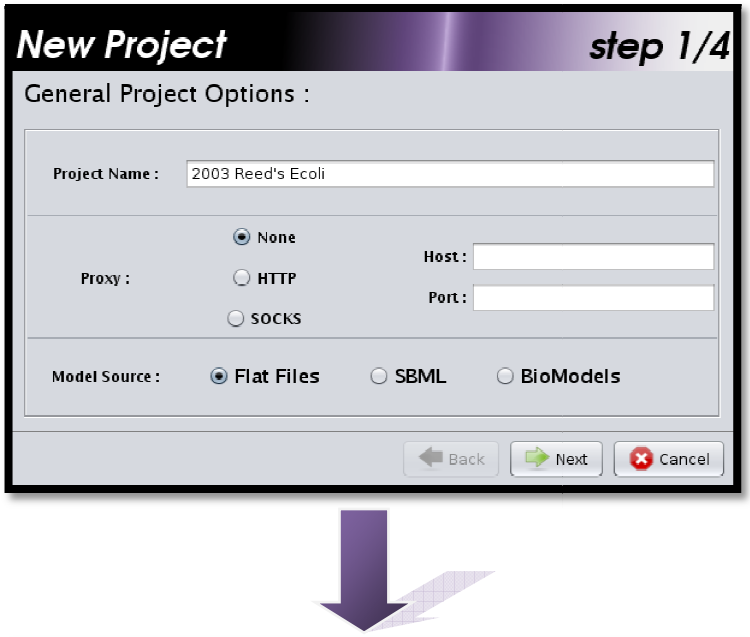


This will launch the **New Project Wizard** step 1 panel. In *OptFlux* a metabolic model can be loaded from 3 difference sources. This choice is made early in the wizard and affects the rest of its behaviour.

- **FROM LOCAL FLAT FILES:**

Step 1

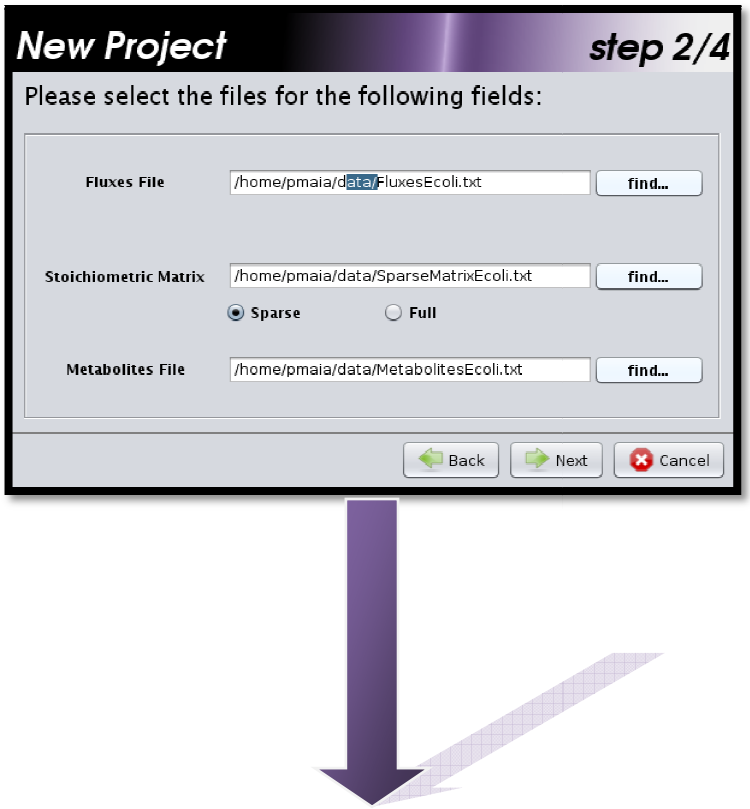
In the first step the user must input a valid project name (anything different from white spaces or tabs). No proxy information is necessary in this step.



Step 2

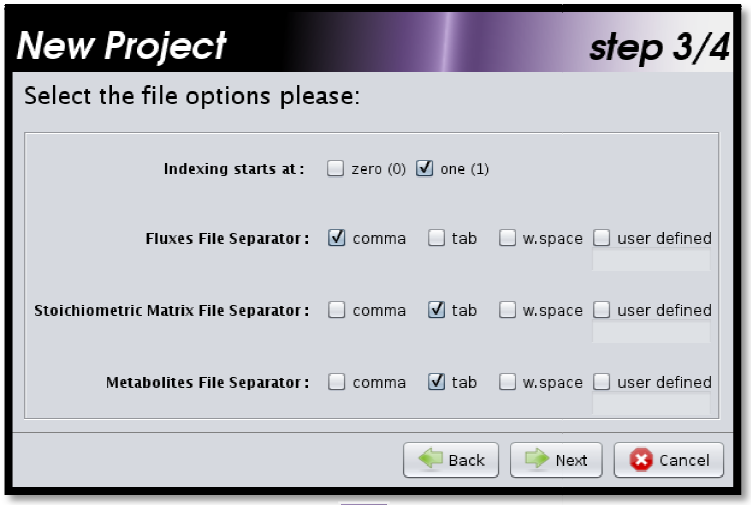
In the second step the user must select three files.

- The first contains the reactions names and their flux limits;
- The second contains the stoichiometric coefficients;
- The third contains the metabolite names (optional)



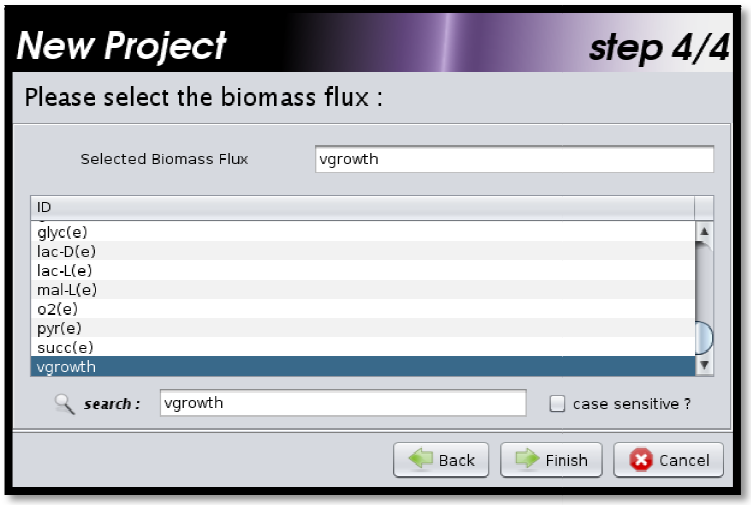
Step 3

In step 3, the first option concerns the indexing used in the stoichiometric matrix if the SPARSE option was selected.
The rest of the options are relative to each of the three files selected previously. For each one, the user should select the appropriate separator.


New Project
step 3/4
Select the file options please:
Indexing starts at : zero (0) one (1)
Fluxes File Separator : comma tab w.space user defined
Stoichiometric Matrix File Separator : comma tab w.space user defined
Metabolites File Separator : comma tab w.space user defined
Back
Next
Cancel

Step 4

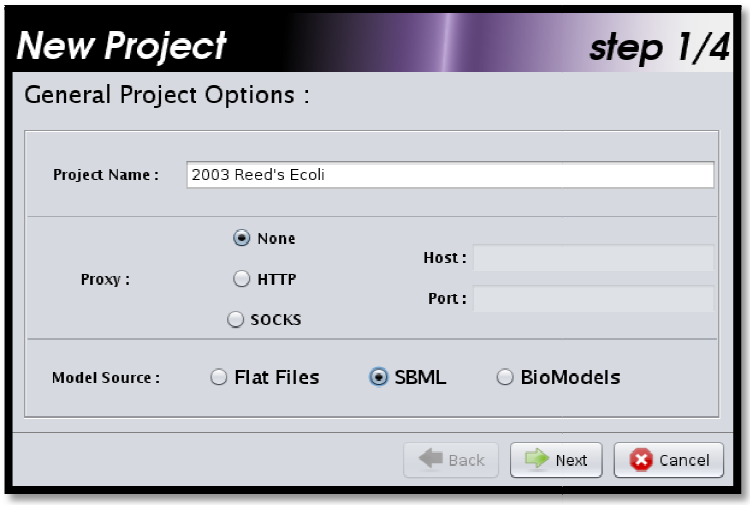
In the fourth step, OptFlux automatically tries to find the biomass growth associated flux, since this information is essential for both simulation and optimization procedures.
If the heuristic fails to detect the correct flux, the user should manually select it.



- **FROM A LOCAL SBML MODEL:**

Step 1

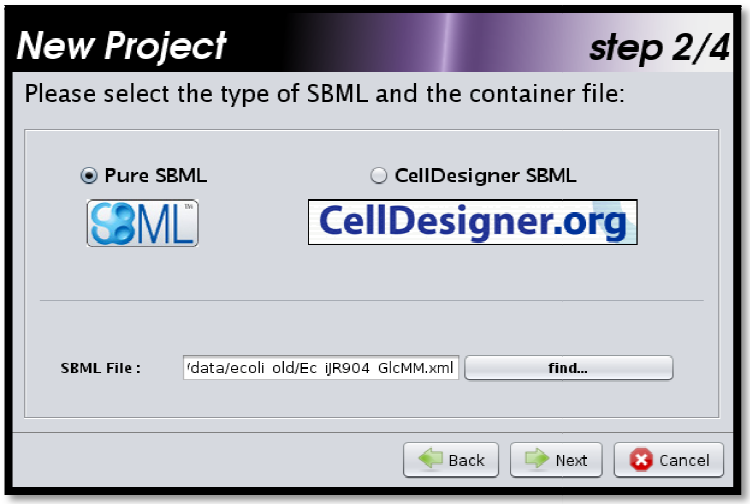
In the first step the user must input a valid project name (anything different from white spaces or tabs). No proxy information is necessary in this step. The user must select the SBML option in the bottom.



Step 2

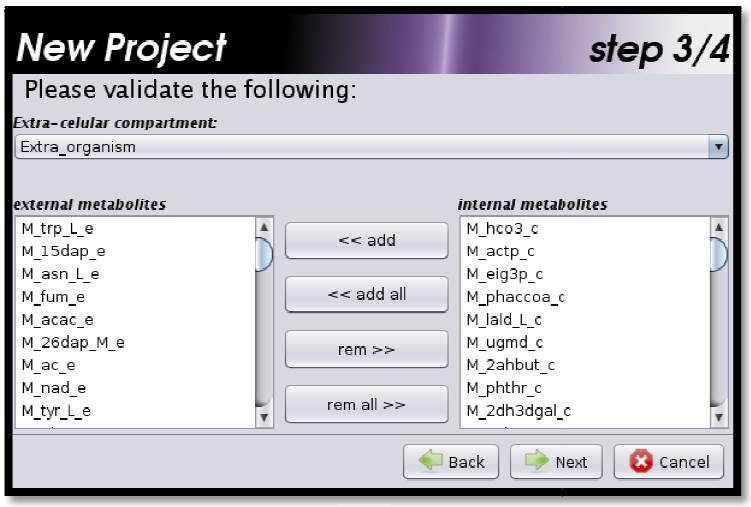
In the second step the user must select the file to load and the type of SBML therein contained, either:

- **Pure SBML**. Normal SBML files (level 2 version 3 supported)
- **CellDesigner SBML**. CellDesigner's generated files, containing layout and visual information. (version 4.0). If this option is selected, OptFlux will try to generate a visualization graph, representing the model. Large models (genome-scale) are very hard to render and, to prevent that, OptFlux will not create the graph if a model with +200 metabolites is loaded.



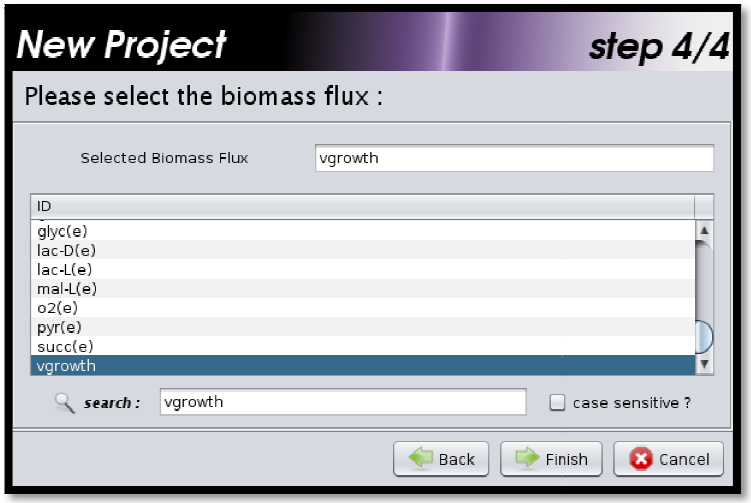
Step 3

The third step is relative to the extra-cellular environment. *OptFlux* will automatically try to find the extra-cellular compartment and the respective metabolites.
This heuristic can, however, fail. In that event, the user must manually select and validate this information.



Step 4

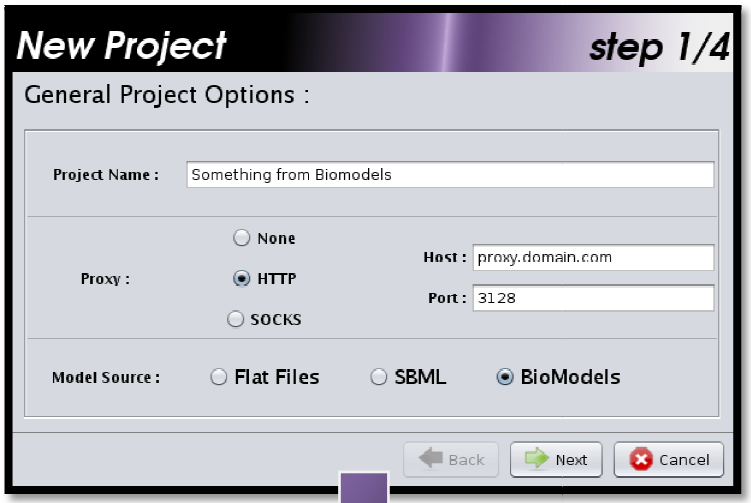
In the fourth step, OptFlux automatically tries to find the biomass growth associated flux, since this information is essential for both simulation and optimization procedures.
If the heuristic fails to detect the correct flux, the user should manually select it.



- **FROM THE REMOTE BIOMODELS REPOSITORY.**

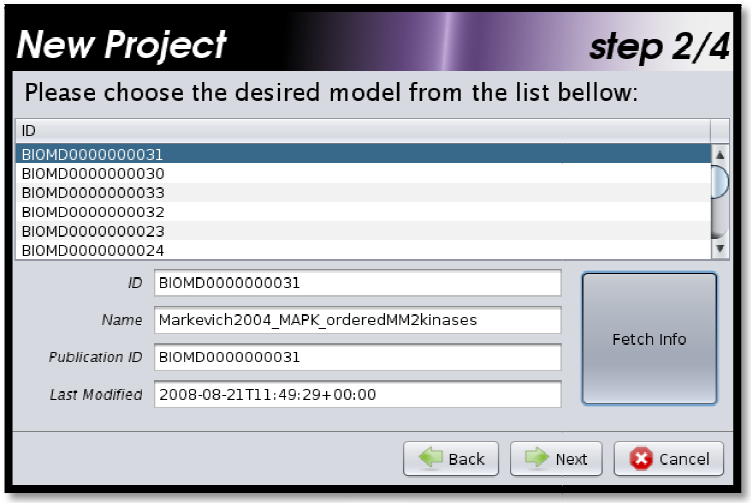
Step 1

In the first step the user must input a valid project name (anything different from white spaces or tabs). When the user computer is behind a proxy, that information must be provided to *OptFlux*, in order to grant remote access.

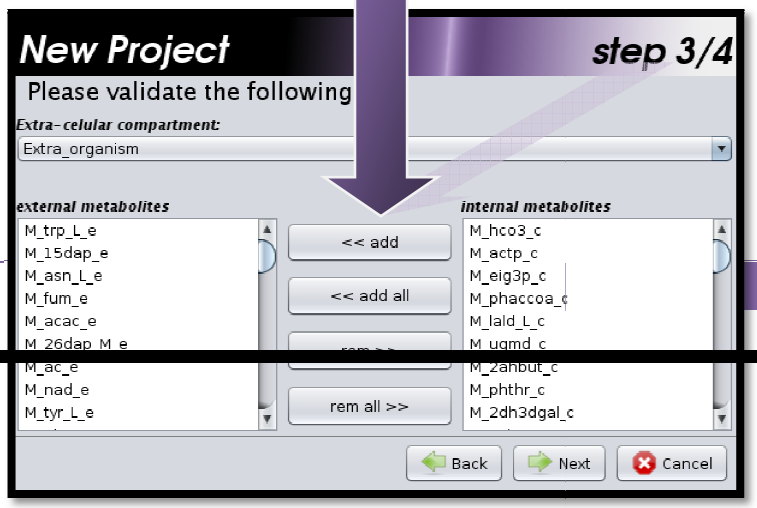


Step 2

In the second step the user must select the file which model to load. The above list shows only the model ID, but when the *"Fetch Info"* button is pressed, a more complete description of the selected model is provided.



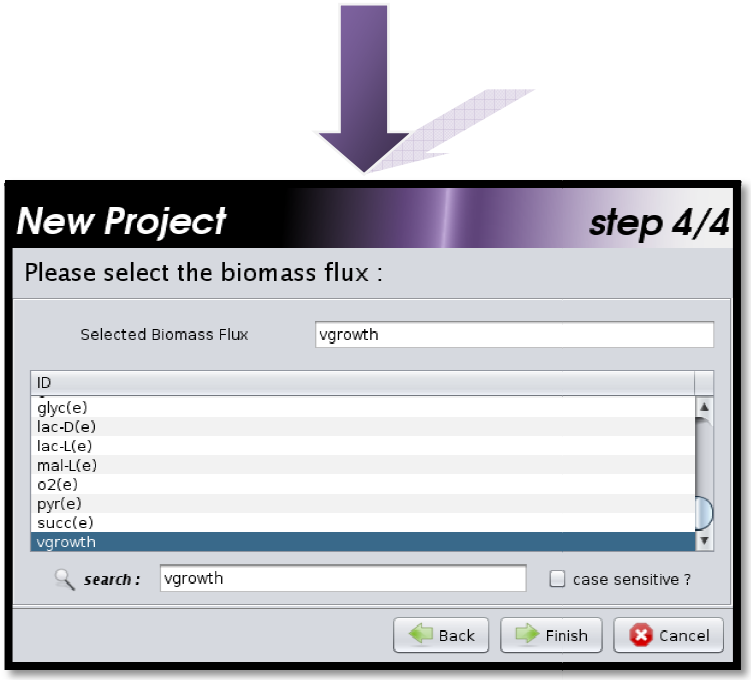
Step 3



The third step is relative to the extra-cellular environment. *OptFlux* will automatically try to find the extra-cellular compartment and the respective metabolites.
This heuristic can, however, fail. In that event, the user must manually select and validate this information.

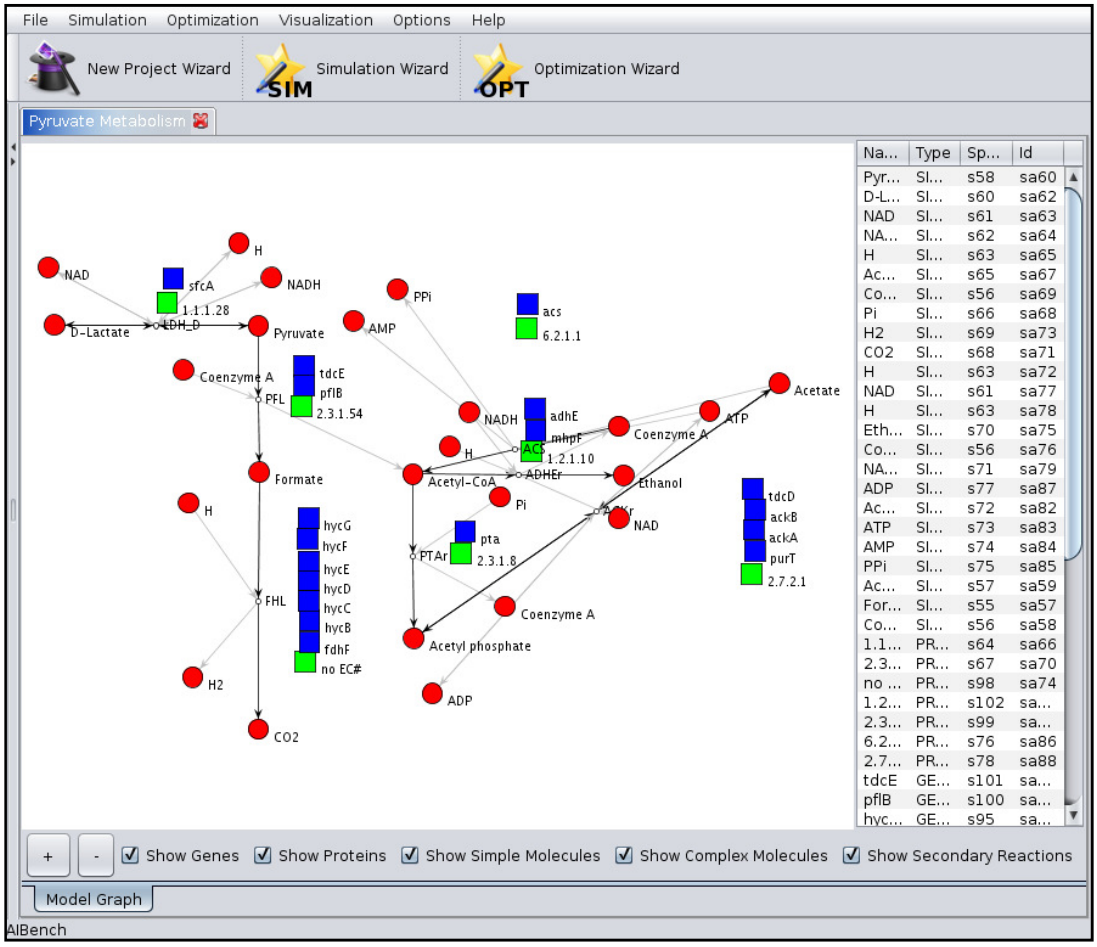
Step 4

In the fourth step, OptFlux automatically tries to find the biomass growth associated flux, since this information is essential for both simulation and optimization procedures.
If the heuristic fails to detect the correct flux, the user should manually select it.



3.2 – MODEL GRAPH VISUALIZATION

The application is integrated with a complementary tool that allows model visualization. Metabolic models are represented as graphs with different kinds of nodes (e.g. genes, enzymes, metabolites). An SBML file can be loaded (option File/ Load SBML for Visualization) to construct the graph, if the model has not been created directly from a SBML file. This will create a data object of the type **ModelGraph**.



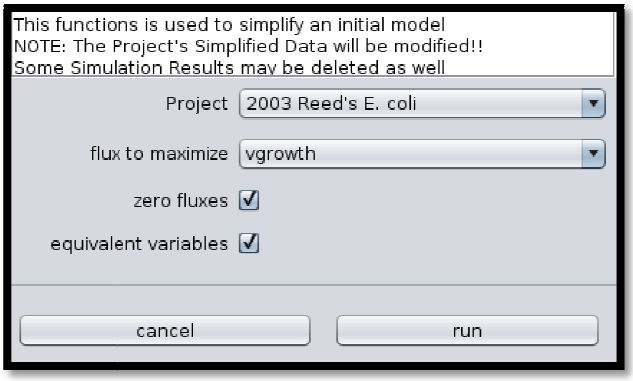
- ***simple molecules***
- ***proteins***
- ***ions***
- ***genes***
- ***prim. reactions***
- ***sec. reactions***

The integration of the SBML visualization graph and the metabolic model objects is done using the names of the reactions/fluxes. Therefore, the visualization graph can be related to the whole set of fluxes or only to a subset.

3.3 – SIMPLIFYING MODELS

In the *Optimization menu*, *Preprocessing sub-menu*, the operation "Model Simplification" allows to choose among two types of simplification procedures (or both):

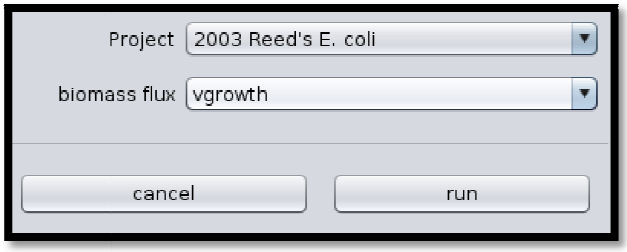
- Remove variables (reactions) that are constrained by the metabolic model to have flux values of 0;
- Identify groups of equivalent variables and replace these groups by a single variable.



These operations create a new metabolic model (denoted as simplified model) and also produce lists of the identified variables (kept in data objects named Equivalent Variables and Zero Valued Variables). It should be noticed that the identification of zero value variables is time demanding.

3.4 – IDENTIFYING ESSENTIAL GENES

This operation (also in the *Preprocessing sub-menu*) allows the identification of genes that when knocked out lead to a biomass reaction flux of near 0. Therefore, the corresponding reactions cannot be removed from the metabolism in order to keep the organism biologically viable.

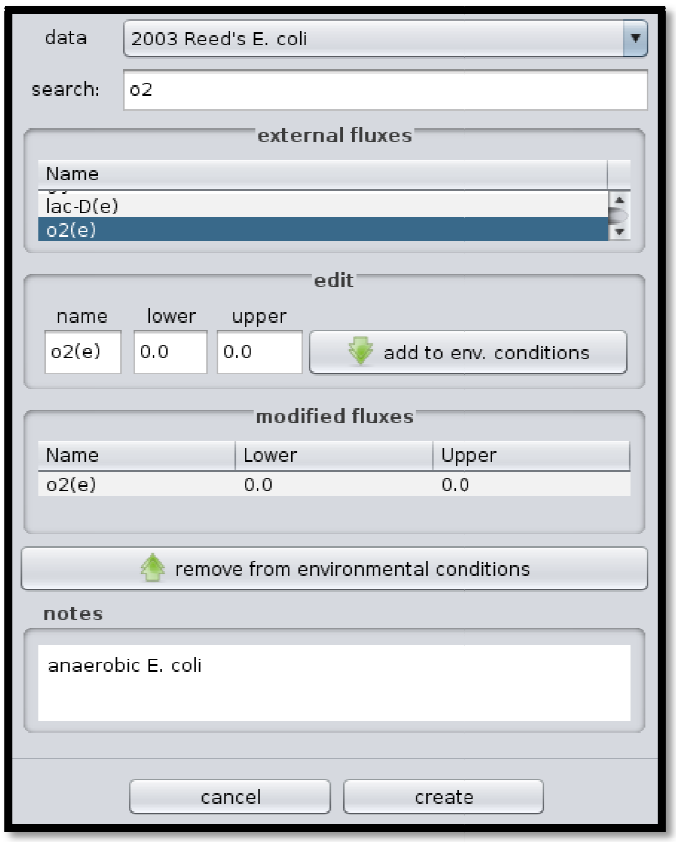


The list of genes that is created in this option can be used in the optimization

algorithms in order to limit the search space. This list can be manually edited using the visualization of the corresponding data object (Essential Genes).

3.5 – USING ENVIRONMENTAL CONDITIONS

The Environmental Conditions are a mechanism developed, to ensure that a researcher has the possibility to change some conditions of a problem, leaving the stoichiometric matrix of the original model untouched. This means that the user can change the limits of certain fluxes, creating a data object (**EnvironmentalConditions**) that can later be used either in simulations or optimizations.



These fluxes are, by default, limited to the external fluxes, the ones that represent the environmental conditions themselves (availability of carbon source and oxygen, for example). This restriction can be, however, overcome to grant the possibility to change the limits of internal fluxes as well (see 4.10).

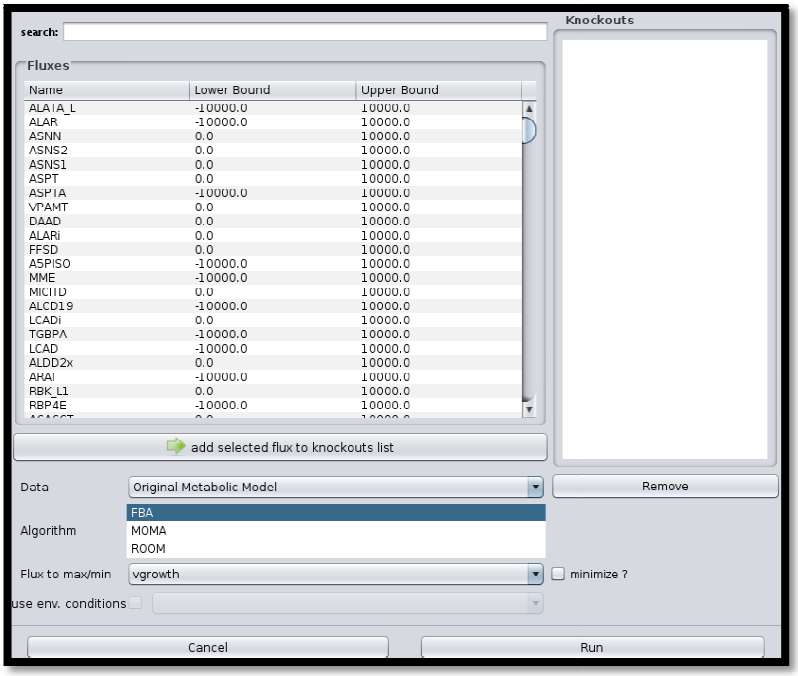
When the data object **EnvironmentalConditions** has been created, the field *Use Environmental Conditions* appears as selectable in simulation and optimization processes, and the user can choose between the available instances and use them in the simulation/optimization. This allows the observation of the behaviour of the organism when the original conditions are replaced by the chosen ones, while keeping the original stoichiometric matrix model safely unmodified.

3.6 – SIMULATION

The simulation menu operations allow the user to test the behaviour of both wild type and mutant strains under different conditions. The simulation of mutants is possible by selecting the reactions that should be eliminated from the metabolic model from the complete list of reactions. The environmental conditions can be defined by appropriately setting the limits of the fluxes as described in the previous section. The result of the simulation will consist on the numerical values for the complete set of the fluxes of the metabolic model.

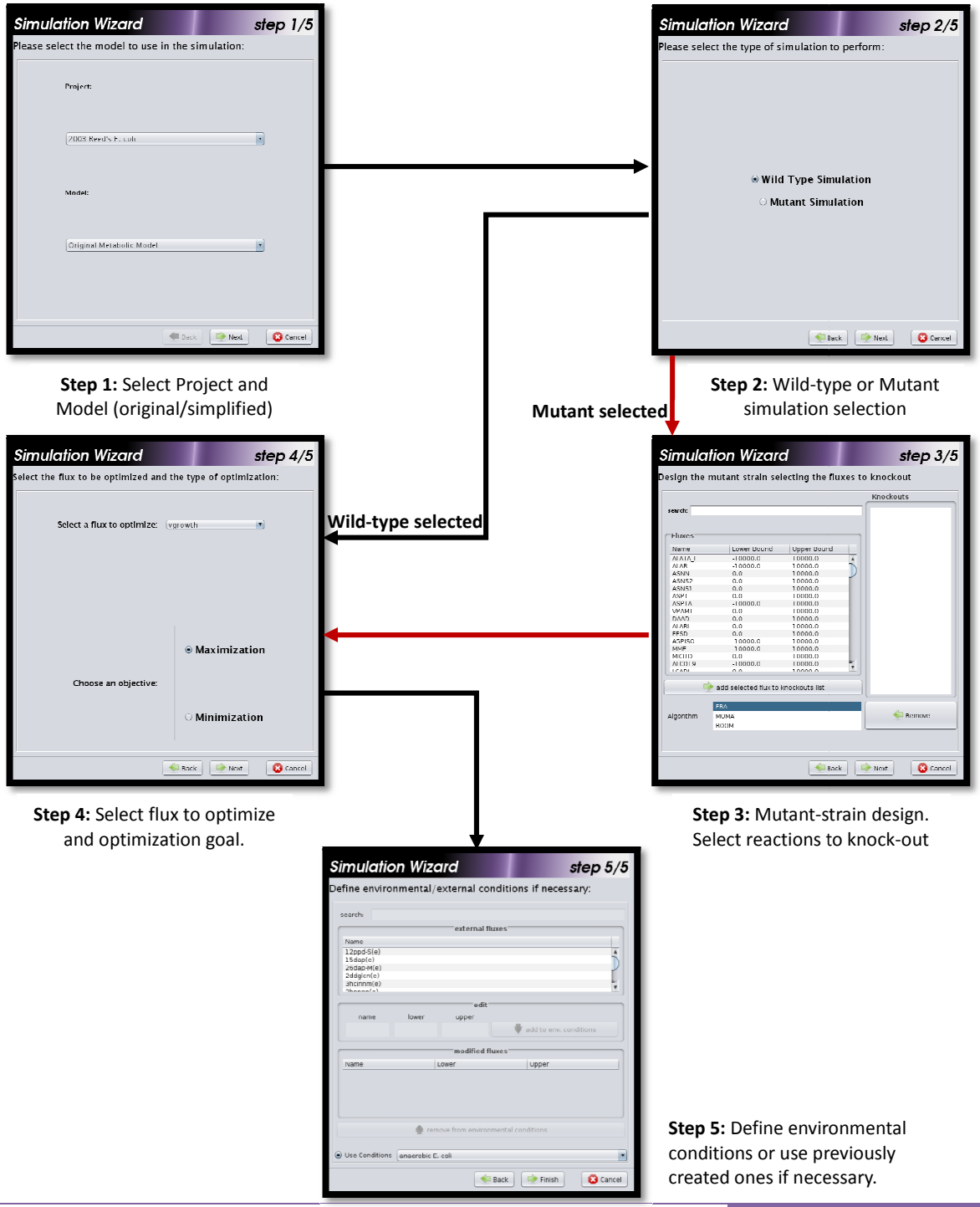
In this version, the simulation methods included are Flux Balance Analysis (FBA), Minimization of Metabolic Adjustment (MoMA) and the Regulatory On/Off Minimization (ROOM). In FBA, a linear programming problem is solved, where one flux, typically representing biomass production, is maximized. In MoMA, wild types follow the same strategy as with FBA, while for mutants, a quadratic problem is defined, where the purpose is to minimize the differences of the fluxes to the wild type solution. The implementation of ROOM was based on a LP relaxation of the original MILP formulation (proposed by the original authors). The interface allows the user to define the flux to be maximized in the linear programming problem (typically the flux over the reaction representing biomass formation).

The simulation operations create data objects of type Simulation Solution as a result. These objects have two viewers: one allows checking the values obtained for all the fluxes in the model after the simulation. The other allows these values to be superimposed over the model graph, using the *BioVisualizer* options. Both viewers can be selected by different tabs.



3.6.1 – USING THE SIMULATION WIZARD

A wizard for simulation is also available in the software. The user can access it both through the *Simulation Menu* or the *Toolbar*. The general steps of the process are the following:



Step 1: Select Project and Model (original/simplified)

Step 2: Wild-type or Mutant simulation selection

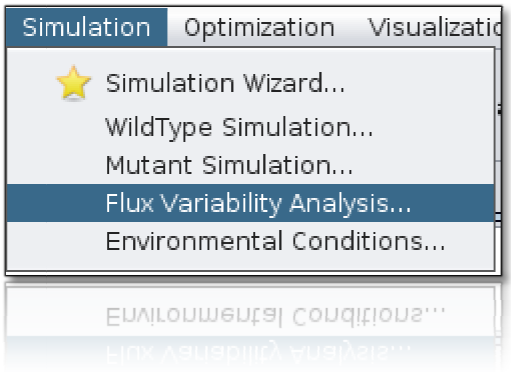
Step 3: Mutant-strain design. Select reactions to knock-out

Step 4: Select flux to optimize and optimization goal.

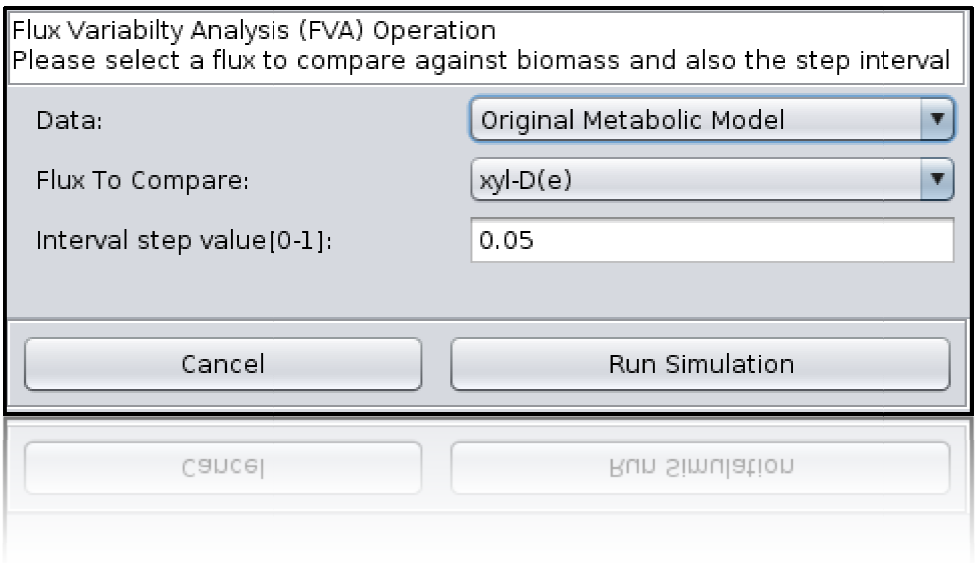
Step 5: Define environmental conditions or use previously created ones if necessary.

3.6.2 – FLUX VARIABILITY ANALYSIS (FVA)

An additional feature in OptFlux is the Flux Variability Analysis (FVA) that allows the user to examine the maximum possible value of a selected flux, for a range of values for the biomass (typically varying from 0 to 100% of its value in the wild type strain).



Users can launch this operation by accessing the *Simulation Menu* as depicted in the picture above. This will launch the operation panel:



The selected interval step must be between 0 and 1. Users should take into account that the smaller the step size, the longer this operation will take. Results will be kept inside the *Variability Analysis Results* list in the Clipboard.

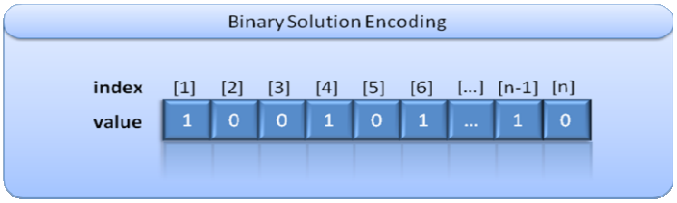
3.7 – Optimization with Evolutionary Algorithms

Evolutionary Algorithms (EAs) are optimization algorithms based on an analogy with natural evolution.

In this application, there are two variants of EAs, namely a Set-based Representation Evolutionary Algorithm (SBREA) and traditional binary based Genetic Algorithms (GA).

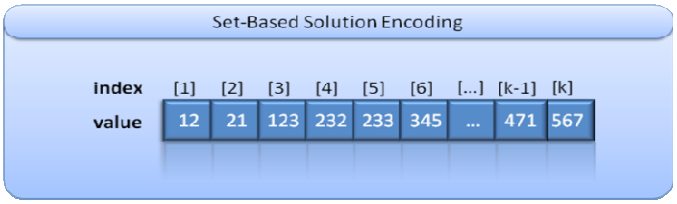
3.7.1 - Solutions Encoding

When using the traditional binary based representation, the solutions will have the size of the complete genome, i.e., if the model that the user is using has n fluxes, the solutions length will be also n.



The above image shows how a solution is encoded in the binary representation. The solution has *n* indexes, the same as the number of fluxes in the model. For any given index, the corresponding value tells the GA if that gene is prone to be knocked-out (value = 0) or not (value =1).

On the other hand, the set-based representation, allows for the definition of shorter solutions. The solution will have as many indexes as the number of knockouts (*k*).

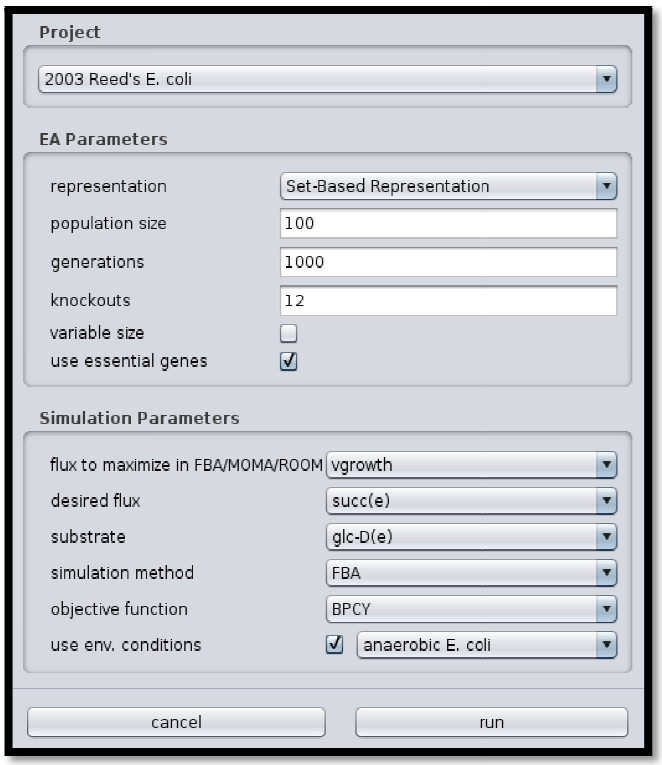


The image above shows a typical instance of a set-based encoded solution. The values corresponding to the indexes, are the indexes of the fluxes (in the original model) that should be knocked out. It is obvious that a solution of this type will have much smaller solutions, allowing the execution of the EA to be less time and memory consuming.

The aim is to obtain modified microbial strains with better production capabilities that

are evaluated by their fitness function. For each potential solution (a given mutant strain) this function is calculated by performing a simulation using the FBA, MOMA or ROOM approaches, as described above.

In this version, the BPCY (Biomass-Product Coupled Yield) is used as a fitness function. BPCY is an objective function that allows simultaneous maximization of both product and biomass and that can be associated with the productivity of a given bioprocess[4].



The interface allows the user to define the fluxes to be maximized, both by the FBA, MOMA or ROOM (typically the flux leading to biomass) and by the EA optimization process (the flux leading to the target product used in the calculation of the BPCY- e.g. succinate or ethanol), as well as the substrate uptake flux (corresponding to the main carbon source, whose value is needed for the calculation of the fitness function BPCY).

The user can also define a few EA parameters: the number of generations and the EA's population size. The multiplication of the values of these two parameters gives an approximation of the number of solutions that will be evaluated during the search process.
The optimization procedure using EAs is typically time consuming.

In SBREAs, an important feature is the possibility of evolving solutions with variable number of knockouts, meaning that the user does not have to define *a priori* the number of genes to be deleted (information that is usually not known). Additionally, it is possible to define an upper limit on the number of genes to be deleted, since solutions with a large number of knockouts are difficult to implement in the lab.

The output of the EA is a set of processed solutions that result from the application of

a simplifying process to the raw solutions obtained by the EA. This process removes all the repeated solutions and removes from each solution the knockouts that are not strictly necessary to obtain a good fitness value.

The solutions resulting from this process can be kept in an object of the type "Solution Set" and further analysed. This object is essentially a list of objects of the type Simulation Solution mentioned before, ordered by the value of objective function. Thus, each solution in the list can be selected and analysed using the tools mentioned in the previous section.

3.8 – Optimization with Simulated Annealing

Simulated Annealing (SA) is an optimization algorithm inspired in the annealing process used in metallurgy, where a melt, initially at high temperature, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium.

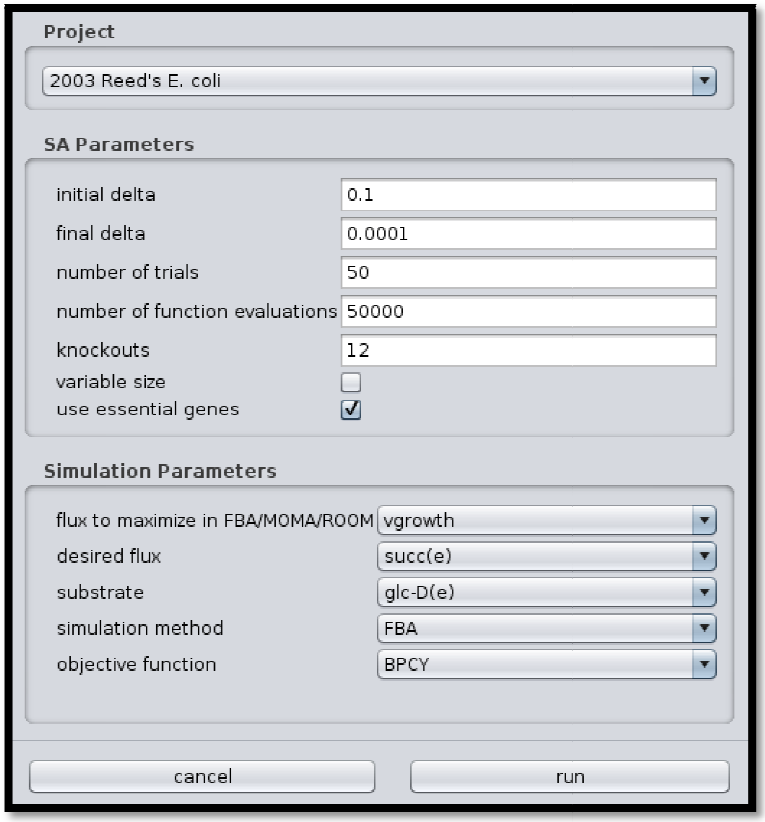
As the cooling proceeds, the system becomes more ordered, approaching a minimal energy state when the temperature reaches zero. If the initial temperature of the system is too low or the cooling process is not sufficiently slow the system may become trapped in a local minimum energy state.
In the original Metropolis scheme, an initial state of a thermodynamic system is chosen at energy E and holding temperature T constant. The initial configuration is perturbed and the change in energy ΔE is computed. A better configuration is always accepted, while a worse configuration is only accepted with a probability given by the Boltzmann factor:

$$p[accept] = e^{-\frac{\Delta E}{T}}$$

The Interface allows the user to define several different parameters.

Similarly to the EA interface, it allows the user to define the fluxes to be maximized, both by the FBA or MoMA and by the SA optimization process, as well as the substrate uptake flux.

Furthermore, the interface allows the definition of some of the parameters of the SA itself. The configuration parameters for the algorithm are the initial and final temperatures, the number of iterations performed at each temperature and the total number of function evaluations.



The choice of these parameters is of paramount importance to the performance of the algorithm. If the initial temperature is too low or the cooling schedule is not slow enough, the optimization process may become stuck in a local optimum. On the other hand, if the initial temperature is too high, the cooling is too slow or the number of iterations per temperature is too high, the algorithm wastes a potentially large amount of computational time while searching for solutions.

The cooling schedule used by *OptFlux* is among the most popular ones, where the temperature decreases exponentially, defined according to the following equation:

To ensure that the cooling schedule is sufficiently slow, the parameter $\boldsymbol{\alpha}$ should be given values close to the unity. As the choice of initial ($\boldsymbol{T_0}$) and final temperatures ($\boldsymbol{T_f}$) is problem dependent, it was decided to use as configuration parameters the following:

- $\boldsymbol{\Delta E_0}$ ***(initial delta)*** – The difference in energy that corresponds to an acceptance probability of 50% of worse solutions at the beginning of the run;
- $\boldsymbol{\Delta E_f}$ ***(final delta)*** – The difference in energy that corresponds to an acceptance

probability of 50% of worse solutions at the end of the run;

- ***trials*** – The number of iterations per temperature;
- ***NFEs*** – The number of function evaluations.

Using these parameters, the initial temperature, the final temperature and the scale parameter are computed by *OptFlux* using the following equations:

$$T_0 = -\frac{\Delta E_0}{\log 0.5}$$

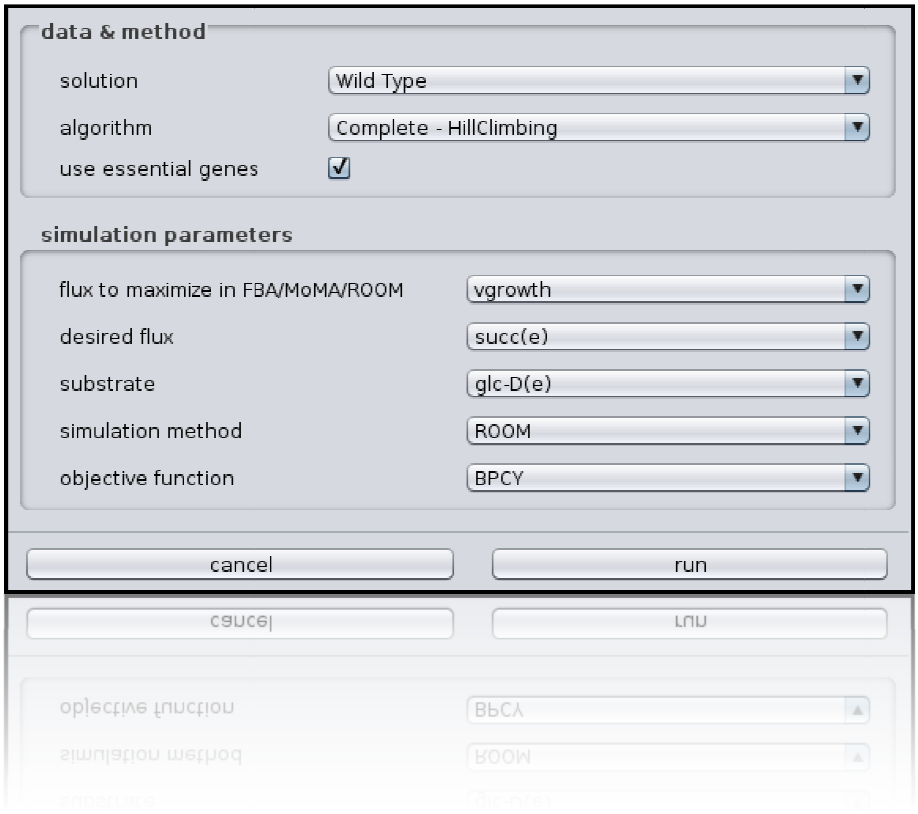
$$T_f = -\frac{\Delta E_f}{\log 0.5}$$

$$\alpha = \exp\left(\frac{\log \mathrm{T_f} - \log \mathrm{T_0}}{\left[\frac{\mathrm{NFEs}}{\mathrm{trials}}\right]}\right)$$

The advantage of using $\boldsymbol{\Delta E_0}$ and $\boldsymbol{\Delta E_f}$ is that it allows the user who knows the fitness landscape of the optimization problem to automatically define the temperatures by reasoning over the values of the objective function. Supplying the number of function evaluations instead of the scale parameter allows the user to accurately define the number of function evaluations the optimization algorithm will use, enabling a simpler comparison with other approaches.

3.9 – OPTIMIZATION WITH LOCAL SEARCH

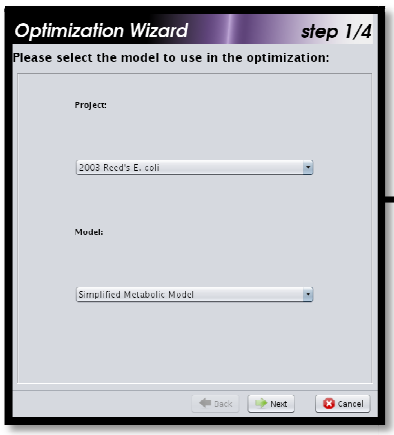
Local search algorithms are used in *OptFlux* to improve a particular solution. Therefore, this operation can be applied to any existing solution (wild type or mutant). Basically, the local search algorithm tries to add a gene deletion to an existing solution trying to improve the objective function. All possible genes are attempted and the best solution is returned.



This cycle can be executed once (Best-Neighbour), or repeated while it provides an improved solution over the starting point (HillClimbing). If an improved solution is found, the best solution is added to the clipboard.

3.10 – USING THE OPTIMIZATION WIZARD

A wizard for optimization is also available in the software. The user can access it both through the *Simulation Menu* or the *Toolbar*. The general steps of the process are the following:



Step 1: Select Project and Model (original/simplified)

Step 2: Select EA's or SA's as optimization algorithm

SA's selected

EA's selected

Step 3: Define simulation and optimization parameters for the EA algorithm.

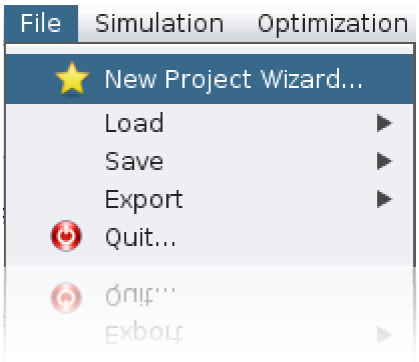
Step 3: Define simulation and optimization parameters for the SA algorithm.

Step 4: Define environmental conditions or use previously created ones if necessary.

4 | Operations – Complete Reference

In this chapter, a complete reference on the operations of *OptFlux* is presented. The structure will follow the structure of the menus.

4.1 – File



4.1.1 – New Project Wizard

Provides a step-by-step wizard to create a new Project in the Clipboard. This is always the first operation to be executed in *OptFlux*, otherwise, no other will work.

- **STEP1**

Project Name	The name of the project. Name validation will be performed.	
Proxy	The type of proxy to bypass if necessary. Either **NONE**, **HTTP** or **SOCKS**.	
	Host	The proxy host if **HTTP** or **SOCKS** is selected.
	Port	The proxy port if **HTTP** or **SOCKS** is selected-
Model Source	The source of the model to be loaded. Either **Flat Files (Source1)**, **SBML (Source2)** or **BioModels (Source3)**. This option will affect the behavior of the wizard for the following steps and, therefore, the following steps will be described with **Source1**, **Source2** or **Source3** depending on the context they apply to.	

- **STEP2**

(SOURCE1)

Fluxes File	The file containing the flux bounds for the reactions. For **supported formats**, refer to **7.1.1**.	
Stoichiometric Matrix	The file containing the matrix with the stoichiometric coefficients.	
	SPARSE	Select this option if the loaded matrix is in sparse format. Refer to **7.1.2** for description of the sparse format.
	FULL	Select this option if the loaded matrix is in its full format. Refer to **7.1.3** for description of the full format.
Metabolites File	The file containing the metabolites information. For the format of this file, please refer to **7.1.4**.	

(SOURCE2)

	Pure SBML	Select this option if the loaded model is in Pure SBML format (refer to **7.3.1**)
	CellDesigner SBML	Select this option if the loaded model is in CellDesigner's SBML format (refer to **7.3.2**)
SBML File	The SBML file containing the model to be loaded.	

(SOURCE3)

[ID]	The list of ID's provided by the BioModels database. One must be selected to be loaded.
Fetch Info	By pressing this button, *OptFlux* will fetch complementary information for the Model ID selected in the list above. The information will be placed in the fields to the left of this button.

- **STEP3**

(SOURCE1)

Indexing starts at	The number used to identify the first index (used in sparse matrices only), either **0** or **1**.
Fluxes File Separator	The separator used to discriminate elements in this file. Either **Comma**, **Tab**, **White Space** or a **User Defined Separator**
Stoichiometric Matrix File Separator	The separator used to discriminate elements in this file. Either **Comma**, **Tab**, **White Space** or a **User Defined Separator**

Metabolites File Separator	The separator used to discriminate elements in this file. Either **Comma**, **Tab**, **White Space** or a **User Defined Separator**

(SOURCE2 AND SOURCE3)

Extra-cellular Compartment	The compartment representing the **extra-cellular environment** in the SBML model loaded.

[External Metabolites]	[Internal Metabolites]
The list of detected or manually assigned **external metabolites**. This information is essential for simulation purposes. Sink fluxes will be created if non-existent.	The list of internal metabolites. Users can add or remove metabolites to the external metabolites list, if *OptFlux* heuristics failed to correctly detect them.

- **STEP4**

(SOURCE1, SOURCE2 AND SOURCE 3)

Selected Biomass flux	Displays the currently selected flux that will be assumed by *OptFlux* as the biomass growth associated flux.
[ID, NAME]	The list of fluxes present in the model. The biomass growth associated flux should be selected by default by *OptFlux*, if not, the user must manually select it.
Search	Field used to perform a quick search for the desired flux. Must only be used if *OptFlux* heuristics fail to correctly detect the biomass growth associated flux.

4.1.2 – LOAD

- **LOAD PROJECT:**

  Recovers a Project from a previously saved file.

File	The location of the file where the project was saved to.

- **LOAD SBML FOR VISUALIZATION:**

  Loads a SBML model with a graph representation to be visualized.

Project	The Project to where the SBML model will be loaded.
Model Type	The type of SBML model that will be loaded. Either SBML or CSBML.
File	The file where the SBML model is located.

- **LOAD CRITICAL REACTIONS:**

  Loads a list of critical reactions from file. The loaded reaction must be contained in the model, otherwise an Exception will be thrown.

Project	The Project to where the list of essential genes will be loaded.
BIN or ASCII	The type of file where the list is saved in. Either BIN (binary file) or ASCII (text file).
File	The location of the file itself.

  - **ASCII**: The file format for the critical reactions file can be found in **7.2.1**
  - **BIN**: This file does not have a human readable format. It can be generated inside *OptFlux* using the Critical Reactions Viewer (see **6.2.1**). The user can, afterwards, save the list in a Binary format to ease the loading/parsing times.

4.1.3 – SAVE

- **SAVE PROJECT AS:**

  Saves a Project to file. The file will be saved in a binary format, thus not human readable. A XML representation is being studied for the next version.

Project	The Project that to be saved.
File	The file to where the project will be saved.

4.1.4 – EXPORT

- **EXPORT FLUXES:**

  Export the list of fluxes to a text file.

FBAFluxes	The FBAFluxes instance to be exported to text file.
File	The file to where the fluxes will be saved.

- **EXPORT MATRIX:**

  Export the matrix to a text file. The matrix will be saved in a SPARSE format either it was loaded as SPARSE or FULL matrix.

Matrix	The FBASparseMatrix instance to export to text file.
File	The file to where the matrix will be saved.

- **EXPORT METABOLITES:**

  Export the list of metabolites to a text file.

Metabolites	The FBAMetabolites instance to be exported to text file.
File	The file to where the metabolites list will be saved.

- **EXPORT ENVIRONMENTAL CONDITIONS:**

  Export the environmental conditions to a text file.

Env.Conditions	The EnvironmentalConditions instance to be exported to text file.
File	The file to where the Environmental Conditions will be saved.

- **EXPORT COMPLETE MODEL:**

  Exports the complete model to text files. A prefix for the files must be provided as well as a directory to contain them.

Model	The DataBox instance that to be saved to text files. NOTE: three separate files will be created, concerning the fluxes, the matrix and the metabolites.
Path	The location where the files should be created. Note: A folder must be selected, not a file.
Prefix	The prefix that the files will have. The complete names will be PREFIXFluxes.txt, PREFIXMatrix.txt, PREFIXMetabolites.txt.

- **EXPORT CRITICAL REACTIONS:**

  Export the list of critical reactions to a text file.

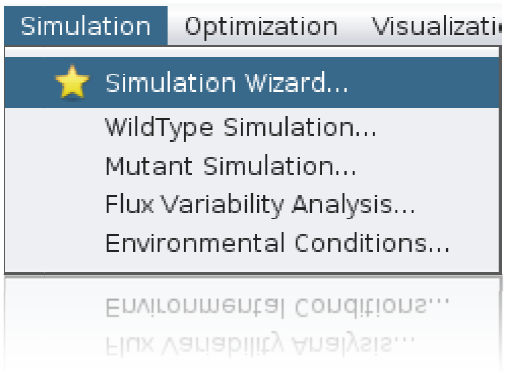
CriticalBox	The CriticalBox instance to be exported to text file.
File	The file to where the critical reactions list will be saved.

4.1.5 – QUIT

Quits *OptFlux,* asking for confirmation and saving.

Project	Select the Project(s) that should be saved before quitting and add them to the list bellow.
(list below)	When the user adds a Project with the above button, they will be added to this list. This list represents the Projects that should be saved prior to quit.

4.2 – SIMULATION



4.2.1 – SIMULATION WIZARD

- **STEP1**

Project	Select the Project where the model to be user is contained.
Model	Select the model to use in the simulation; usually the original or the simplified metabolic model.

- **STEP2**

Type of Simulation	Select the type of simulation to perform; either a **Wild-Type (Option1)** simulation or a Mutant Strain **(Option2)** simulation.

- **STEP3 (OPTION2 ONLY)**

Search	Search for the desired flux.
Fluxes	The list of possible targets for deletion. The user must select the flux to be deleted in the simulation.
Add selected flux to knockouts list	Pressing the button will add the flux selected in the above list to the knockouts list on the right.
Algorithm	The algorithm to use in the simulation. Either FBA(Flux Balance Analysis), MOMA (Minimization Of Metabolic Adjustment) or ROOM (Regulatory On/Off Minimization of metabolic fluxes)
Knockouts	Presents the user with the list of currently select fluxes to be knocked out in this simulation.

- **STEP4 (OPTION1, OPTION2)**

Select a flux to Optimize	The desired flux to optimize in the simulation (usually the biomass-growth associated flux)
Choose an Objective	The optimization objective; either **Maximization** or **Minimization**.

- **STEP5 (OPTION1, OPTION2)**

Search	Search for the desired flux.
external fluxes	The list of possible fluxes. Typically only the external fluxes detected automatically by *OptFlux* will be listed. This can be, however, overridden using Project Preferences with **User Level >= ADVANCED** (see **4.6.1**)
edit	In this panel, the user can edit the value of the flux selected in the above list and add it to the list of modified fluxes.
modified fluxes	The list of fluxes that the user has modified. This list will define the Environmental Conditions themselves. Use the *remove from environmental conditions* button to remove an undesired flux from this list.
Use conditions	Users may choose to use a previously created environmental condition.

4.2.2 – WILD TYPE SIMULATION

Simulation of the Wild-Type of the organism.

Data	The Model to use in this simulation.
biomass flux	The flux to be maximized in the LP problem.
use env. conditions	The Environmental Conditions (if any) to use.

4.2.3 – MUTANT SIMULATION

Simulation of mutant strains.

Search	AYT Search[2]. Search for the desired flux.
Fluxes	The list of possible targets for deletion. The user must select the flux to be deleted in the simulation.
Add selected flux to knockouts list	Pressing the button will add the flux selected in the above list to the knockouts list on the right.
Data	The Model to use in this simulation.
Algorithm	The algorithm to use in the simulation. Either FBA(Flux

2 As You Type Search.

	Balance Analysis), MOMA (Minimization Of Metabolic Adjustment) or ROOM (Regulatory On/Off Minimization of metabolic fluxes)
Flux to maximize	The flux to maximize in the LP or MOMA problem.
use env. conditions	The Environmental Conditions (if any) to use.
Knockouts	Presents the user with the list of currently select fluxes to be knocked out in this simulation.

4.2.4 – FLUX VARIABILITY ANALYSIS

Analyzes the maximum possible value of a selected flux, for a range of values for the biomass.

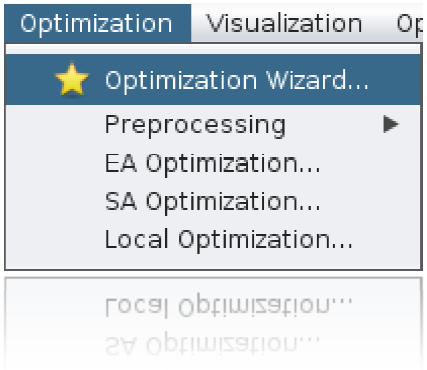
Data	The Model to use in this simulation.
Flux to Compare	Flux to analyze against the biomass growth associated flux.
Interval Step [0-1]	The step size for the biomass to use between simulations. Must be a value between 0 and 1.

4.2.5 – ENVIRONMENTAL CONDITIONS

Interface to define the environmental conditions. These will later be available in simulation and optimization tasks.

Data	Select the Project that contains the desired Model.
Search	AYT Search. Search for the desired flux.
external fluxes	The list of possible fluxes. Typically only the external fluxes detected automatically by *OptFlux* will be listed. This can be, however, overridden using Project Preferences with **User Level >= ADVANCED** (see **4.6.1**)
edit	In this panel, the user can edit the value of the flux selected in the above list and add it to the list of modified fluxes.
modified fluxes	The list of fluxes that the user has modified. This list will define the Environmental Conditions themselves. Use the *remove from environmental conditions* button to remove an undesired flux from this list.
notes	Additional and fully optional comments that the user wishes to add to this Env.Conditions (e.g. Aerobic, Anaerobic, etc.)

4.3 – OPTIMIZATION



4.3.1 – OPTIMIZATION WIZARD

- **STEP1**

Project	Select the Project where the model to be user is contained.
Model	Select the model to use in the simulation; usually the original or the simplified metabolic model.

- **STEP2**

Choose optimization algorithm	Select the Optimization Algorithm to use; either **Evolutionary Algorithm (Option1)** or **Simulated Annealing (Option2).**

- **STEP3**

(OPTION1)

Representation	The desired representation of the individuals to be used by the Evolutionary Algorithm. Either Set-Based Representation or Binary-Based Representation.
population size	The number of individuals that the EA population should have.
generations	The number of generations that the EA should evolve, corresponding to the number of iterations in any classic algorithm.
Knockouts	The number of knockouts that the algorithm must perform.

	Corresponds to the size of the individuals themselves.
variable size	The user must select this to use a variable-sized genome. The algorithm will try to select the optimum size for the individuals.
use essential genes	If a list of essential genes (a CriticalBox instance) exists in the project, the user can use it. The genes contained in that list will never be knocked-out.
flux to maximize in LP/MOMA/ROOM	Select the flux to maximize in the LP/MOMA problem. Usually the biomass flux.
desired flux	Select the flux to maximize in the evolutionary algorithm process. Usually it corresponds to the product that the user wants to maximize the production of.
substrate	Select the flux to use as the substrate (e.g. carbon source, etc.).
simulation method	Select the simulation method used. Either LP or MOMA.
objective function	Select the objective function to use by the EA. Currently only BPCY is available.

(OPTION2)

initial delta	The initial state of energy for the annealing process (see **3.8**).
final delta	The final state of energy for the annealing process (see **3.8**).
number of trials	The number of trials to execute at each temperature(see **3.8**)
number of function evaluations	An approximate maximum number of function evaluations. Allows an easier comparison with other methods (see **3.8**).
knockouts	The number of knockouts that the algorithm must perform. Corresponds to the size of the individuals themselves[3].
variable size	The user must select this to use a variable-sized genome. The algorithm will try to select the optimum size for the individuals.
use essential genes	If a list of essential genes (a CriticalBox instance) exists in the project, the user can use it. The genes contained in that list will never be knocked-out.
flux to maximize in LP/MOMA/ROOM	Select the flux to maximize in the LP/MOMA problem. Usually the biomass flux.
desired flux	Select the flux to maximize in the evolutionary algorithm process. Usually it corresponds to the product that the user wants to maximize the production of.
substrate	Select the flux to use as the substrate (e.g. carbon source, etc.).

3 If the variable size option is selected, this value is only used as the initial size of the individuals.

simulation method	Select the simulation method used; **FBA**, **MOMA** or **ROOM**.
objective function	Select the objective function to use by the EA. Currently only BPCY is available.

- **STEP4 (OPTION1, OPTION2)**

Search	Search for the desired flux.
external fluxes	The list of possible fluxes. Typically only the external fluxes detected automatically by *OptFlux* will be listed. This can be, however, overridden using Project Preferences with **User Level >= ADVANCED** (see **4.6.1**)
edit	In this panel, the user can edit the value of the flux selected in the above list and add it to the list of modified fluxes.
modified fluxes	The list of fluxes that the user has modified. This list will define the Environmental Conditions themselves. Use the *remove from environmental conditions* button to remove an undesired flux from this list.
Use conditions	Users may choose to use a previously created environmental condition.

4.3.2 – PREPROCESSING

- **MODEL SIMPLIFICATION**

Two different options to simplify the model, thus reducing the complexity.

Project	The Project that contains the model to simplify.
Zero Value	Must be selected in order to perform the zero values simplification.
Equivalent Fluxes	Must be selected in order to perform the equivalent fluxes simplification.
Flux to max	The name of the flux to be maximized.

- **FIND CRITICAL REACTIONS**

This operation tries to automatically find the critical reactions in the model, i.e., those that are mathematically prone to be found.

Project	The Project that contains the data.
Flux to max	The name of the flux to be maximized.

4.3.3 – EA OPTIMIZATION

Optimization through Evolutionary Algorithms.

Project	Select the Project containing the desired Model.
representation	The desired representation of the individuals to be used by the Evolutionary Algorithm. Either Set-Based Representation or Binary-Based Representation.
population size	The number of individuals that the EA population should have.
generations	The number of generations that the EA should evolve, corresponding to the number of iterations in any classic algorithm.
Knockouts	The number of knockouts that the algorithm must perform. Corresponds to the size of the individuals themselves[4].
variable size	The user must select this to use a variable-sized genome. The algorithm will try to select the optimum size for the individuals.
use essential genes	If a list of essential genes (a CriticalBox instance) exists in the project, the user can use it. The genes contained in that list will never be knocked-out.
flux to maximize in FBA/MOMA/ROOM	Select the flux to maximize in the FBA/MOMA/ROOM problem. Usually the biomass flux.
desired flux	Select the flux to maximize in the evolutionary algorithm process. Usually it corresponds to the product that the user wants to maximize the production of.
substrate	Select the flux to use as the substrate (e.g. carbon source, etc.).
simulation method	Select the simulation method used. Either FBA/MOMA/ROOM.
objective function	Select the objective function to use by the EA. Currently only BPCY is available.
use env. conditions	The Environmental Conditions (if any) to use.

4 If the variable size option is selected, this value is only used as the initial size of the individuals.

4.3.4 – SA OPTIMIZATION

Optimization through Simulated Annealing.

Project	Select the Project containing the desired Model.
initial delta	The initial state of energy for the annealing process (see **3.8**).
final delta	The final state of energy for the annealing process (see **3.8**).
number of trials	The number of trials to execute at each temperature(see **3.8**)
number of function evaluations	An approximate maximum number of function evaluations. Allows an easier comparison with other methods (see 3.8).
knockouts	The number of knockouts that the algorithm must perform. Corresponds to the size of the individuals themselves[5].
variable size	The user must select this to use a variable-sized genome. The algorithm will try to select the optimum size for the individuals.
use essential genes	If a list of essential genes (a CriticalBox instance) exists in the project, the user can use it. The genes contained in that list will never be knocked-out.
flux to maximize in FBA/MOMA/ROOM	Select the flux to maximize in the LP/MOMA problem. Usually the biomass flux.
desired flux	Select the flux to maximize in the evolutionary algorithm process. Usually it corresponds to the product that the user wants to maximize the production of.
substrate	Select the flux to use as the substrate (e.g. carbon source, etc.).
simulation method	Select the simulation method use; FBA, MOMA or ROOM.
objective function	Select the objective function to use by the EA. Currently only BPCY is available.

4.3.5 – LOCAL OPTIMIZATION

Local optimization of a previously achieved solution. *BestNeighbour* and *HillClimbing* techniques are available.

Solution	Select the solution to improve.

5 If the variable size option is selected, this value is only used as the initial size of the individuals.

algorithm	Select the algorithm to use in the improvement process. Either *BestNeighbour* or *HillClimbing* (see **3.9**).
use essential genes	If a list of critical reactions (a CriticalBox instance) exists in the project, the user can use it. The genes contained in that list will never be knocked-out.
flux to maximize in LP/MOMA/ROOM	Select the flux to maximize in the LP/MOMA problem. Usually the biomass flux.
desired flux	Select the flux to maximize in the evolutionary algorithm process. Usually it corresponds to the product that the user wants to maximize the production of.
substrate	Select the flux to use as the substrate (e.g. carbon source, etc.).
simulation method	Select the simulation method used. Either LP or MOMA.
objective function	Select the objective function to use by the EA. Currently only BPCY is available.

4.4 – OPTIONS

4.4.1 – PROJECT

- **EDIT PREFERENCES**

Edition of some options of the Project. The only available option at the moment is the edition of the user experience and the proxy information. An advanced user will have access to more options inside some operations.

Project	Select the Project whose preferences will be edited.
User Level	Select the user level/experience and proficiency for the selected Project. Either BEGINNER, INTERMEDIATE or ADVANCED.

4.5 – HELP

4.5.1 – ABOUT

Launches an information panel about the coordination and development team of *OptFlux* as well as institutions associated with this project.

5 | DATATYPES – COMPLETE REFERENCE

This chapter presents a complete reference about the Datatypes used in *OptFlux*, listed alphabetically.

5.1 – *AIBENCH* TYPES OF DATATYPES

In *AIBench*, several Datatypes have been defined. There are three major types that give the programmer the ability to define any other Datatype:

Simple	This type of Datatype is the simplest one. It works as a wrapper, for any type of data that the user wishes, but can only contain one instance of a given type of data.
List	This Datatype was created do hold Lists of other Datatypes. Since the objective is for it to hold instances of Lists, it can hold as many objects as the List capacity itself.
Complex	This is the most elaborate Datatype. It was created to act as the glue between all the other Datatypes. This Datatype can hold Simple, List and other Complex Datatypes in it.

5.2 – *OPTFLUX* DATATYPES

5.2.1 – CRITICAL REACTIONS (CRITICALBOX)

Type	Simple
Description	This Datatype holds the list of essential genes. The essential genes are those genes needed to maintain the organism's biomass production.
Viewers	CriticalBox Viewer

5.2.2 – Original Metabolic Model (DataBox)

Type	Complex.	
Description	This Datatype holds the Metabolic Model itself.	
Viewers	Metabolic Model Viewer.	
Contains		
Name	**Description**	**Viewers**
Reactions (FBAFluxes)	Contains the flux names and limits.	Reactions Viewer
Metabolites (FBAMetabolites)	Contains the metabolites information: abbreviation, name, compartment name and compartment location.	Metabolites Viewer
Stoichiometric Model (FBASparseMatrix)	Contains all the stoichiometric coefficients.	Reactions / Steady-State Equations

5.2.3 – Environmental Conditions (EnvironmentalConditionsSet)

Type	List.	
Description	This Datatype holds the list of Environmental Conditions created so far.	
Viewers	NONE. Not Viewable.	
Contains		
Name	**Description**	**Viewers**
[List] EnvironmentalConditions	Each Environmental Conditions define limits value modifications to a set of chosen fluxes.	Environmental Conditions Viewer

5.2.4 – Equivalent Variables (EquivalencesBox)

Type	Simple.
Description	This Datatype holds the Equivalent Variables. Those variables that are constrained in the LP problem to have the same value.

Viewers	Equivalent Variables Viewer

5.2.5 – Model Graph (ModelBox)

Type	Simple.
Description	This Datatype holds the graph information correspondent to the metabolic pathway loaded from a SBML model.
Viewers	Model Graph Viewer.

5.2.6 – OptimizationResults

Type	List.	
Description	This Datatype holds the list of optimization procedures done so far.	
Viewers	NONE. Not Viewable.	
Contains		
Name	**Description**	**Viewers**
[List] SolutionSetBox	Each SolutionSetBox contains a series of solutions for one optimization procedure performed.	Optimization Solutions Viewer

5.2.7 – Project

Type	Complex.	
Description	This Datatype basically holds everything. All the other Datatypes listed in this chapter are in one way or another inside a Project. This is the root Datatype.	
Viewers	NONE. Not Viewable.	
Contains		
Name	**Description**	**Viewers**
Metabolic Model (DataBox)	Holds the metabolic model itself.	Metabolic Model Viewer

Simplified Metabolic Model (SimplifiedDataBox)	The copy of the original metabolic model, after some simplifications.	Metabolic Model Viewer
Simulation Results (SimulationResults)	The list of all the simulations performed so far.	NONE.
Optimization Results (OptimizationResults)	The list of all the optimization procedures performed so far.	NONE.
Environmental Conditions List (EnvironmentalConditionsSet)	The list of environmental conditions created so far.	NONE.
Model Graph (ModelBox)	This Datatype holds the graph information correspondent to the metabolic pathway loaded from a SBML model.	Model Graph Viewer

5.2.8 – Simplified Metabolic Model (SimplifiedDataBox)

Type	Complex.	
Description	This Datatype holds the Metabolic Model itself.	
Viewers	Metabolic Model Viewer.	
Contains		
Name	**Description**	**Viewers**
Reactions (FBAFluxes)	Contains the flux names and limits.	Reactions Viewer
Metabolites (FBAMetabolites)	Contains the metabolites information: abbreviation, name, compartment name and compartment location.	Metabolites Viewer
Stoichiometric Model (FBASparseMatrix)	Contains all the stoichiometric coefficients.	Reactions / Steady-State Equations
Zero Values (ZeroBox)	Contains the list of variables that are constrained to be 0 by the LP problem.	Zero Variables Viewer
Equivalences (EquivalencesBox)	Contains the list of variables that are constrained by the LP problem to have the same value.	Equivalent Variables Viewer

5.2.9 – Simulation Results

Type	List.	
Description	This Datatype holds the list of simulations performed so far.	
Viewers	NONE. Not Viewable.	
Contains		
Name	**Description**	**Viewers**
[List] SimulationSolution	Each SimulationSolution represents a the outcome of a solved LP problem.	SimulationSolution Viewer

5.2.10 – Variability Analysis Results

Type	List.	
Description	This Datatype holds the list of variability analysis performed so far.	
Viewers	NONE. Not Viewable.	
Contains		
Name	**Description**	**Viewers**
[List] FVASimulationResult	Each FVASimulationResult represents the outcome of a variability analysis.	Variability Analysis Viewer

5.2.11 – Zero Fluxes (ZeroBox)

Type	Simple.
Description	This Datatype contains the list of variables that are constrained to be 0 by the LP problem.
Viewers	Zero Variables Viewer

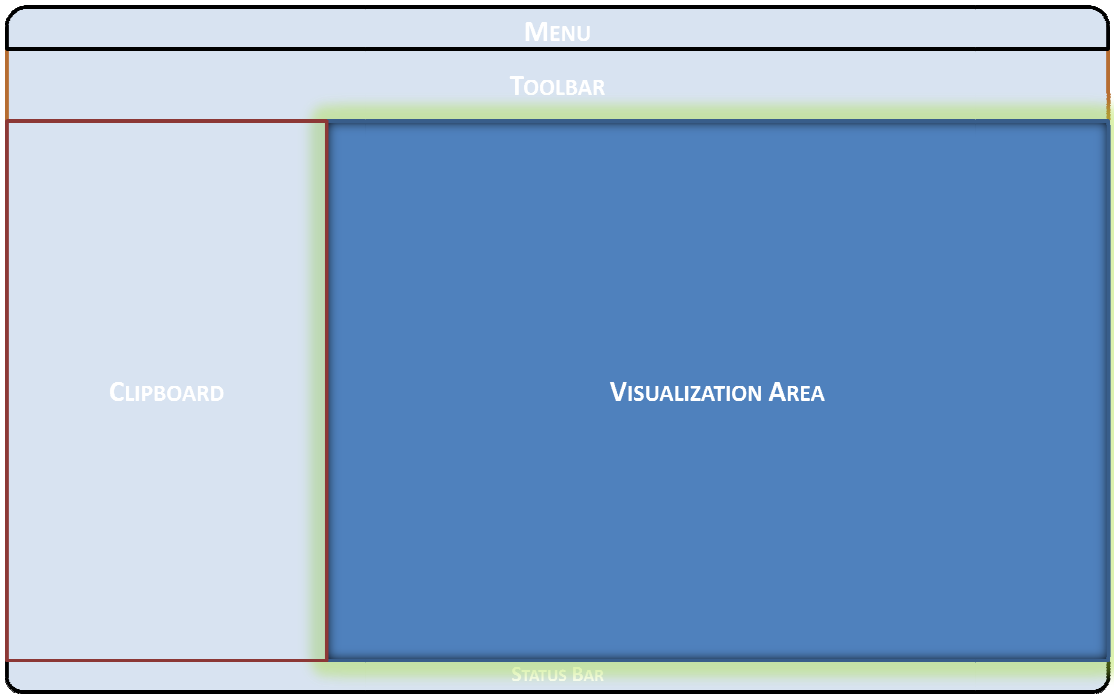
6 | VIEWERS / EDITORS – COMPLETE REFERENCE

In this chapter the Viewers corresponding to the Datatypes presented in the previous chapter will be described and explained, also listed alphabetically.

6.1 – VIEWERS OR EDITORS

Since *OptFlux* follow the MVC software architectural pattern the models (in this case, the data) and the viewers are structurally detached. Nevertheless, it is possible with some viewers to edit data associated with the models and these changes can be passed to the corresponding data. This fact claims for a separate classification for those Viewers that have editing capabilities and those that do not. We will call them Viewers or Viewers/Editors depending on their editing abilities.

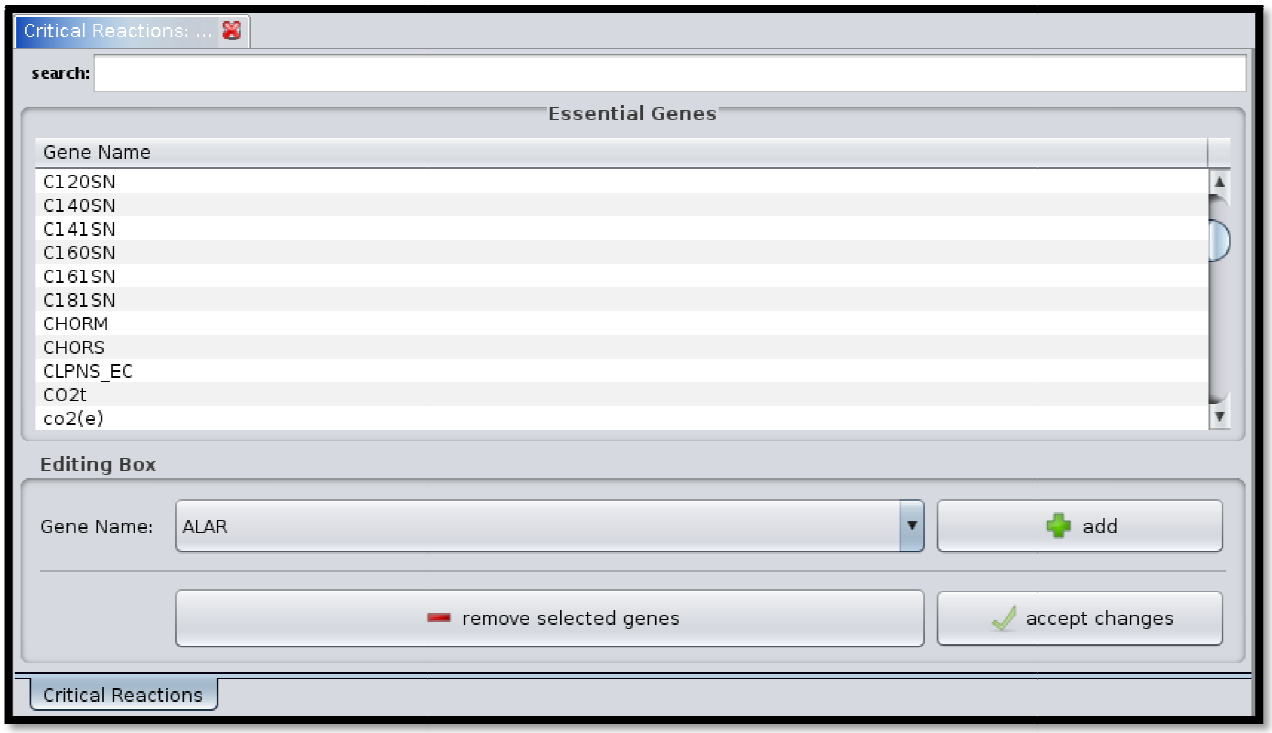
The viewers will always be launched in the visualization area (emphasized in the picture bellow)



6.2 – *OPTFLUX* VIEWERS AND EDITORS

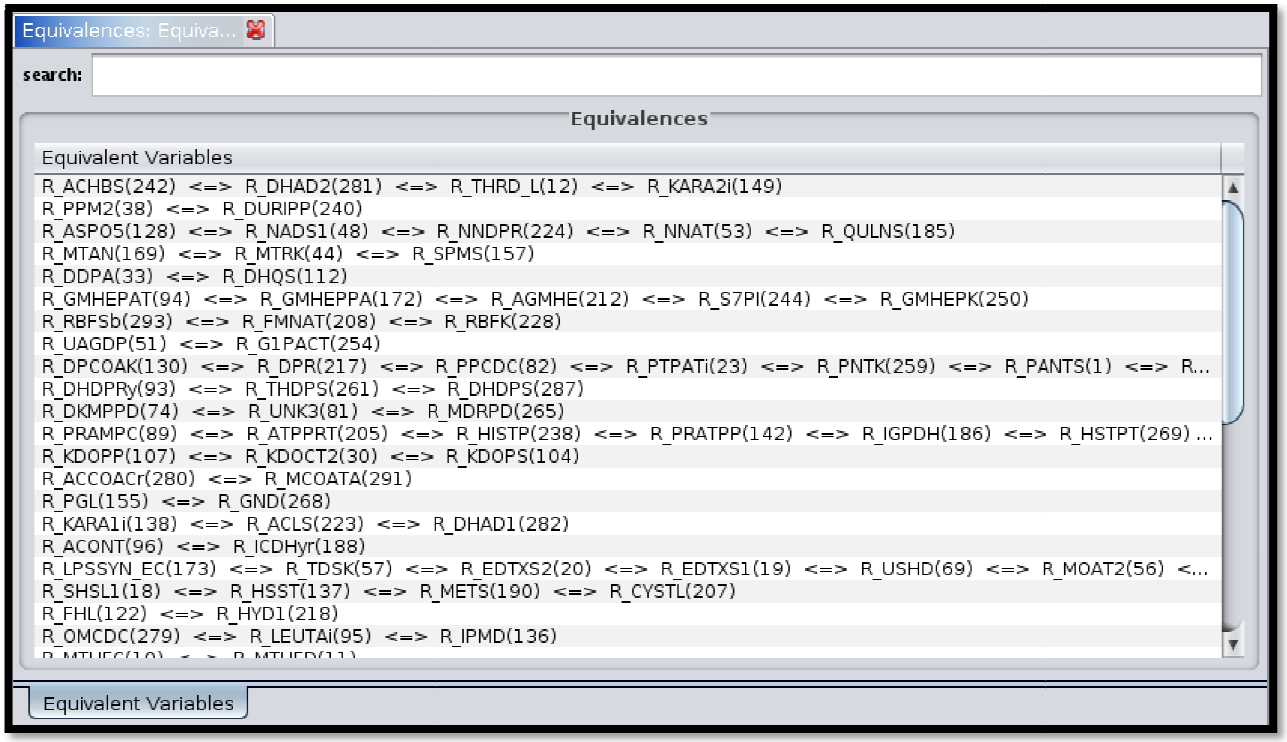
6.2.1 – CRITICAL REACTIONS VIEWER

Name	Critical Reactions Viewer
Type	Viewer / Editor
Description	This Viewer is also an Editor. It allows for the edition of the essential genes list.
Components	
Search	AYT Search. Search for the desired flux in the list.
Essential Genes	The list of essential genes.
Gene Name	The list of genes that aren't in the list yet.
add	Adds the flux selected in the combo to the above list.
remove selected genes	Removes the genes select in the list. The removed genes will be added to the end of the Gene Name combo box.
accept changes	The edition of the essential genes list is done in memory. The changes are only effective when the user press this button and accept them in the posterior dialog.



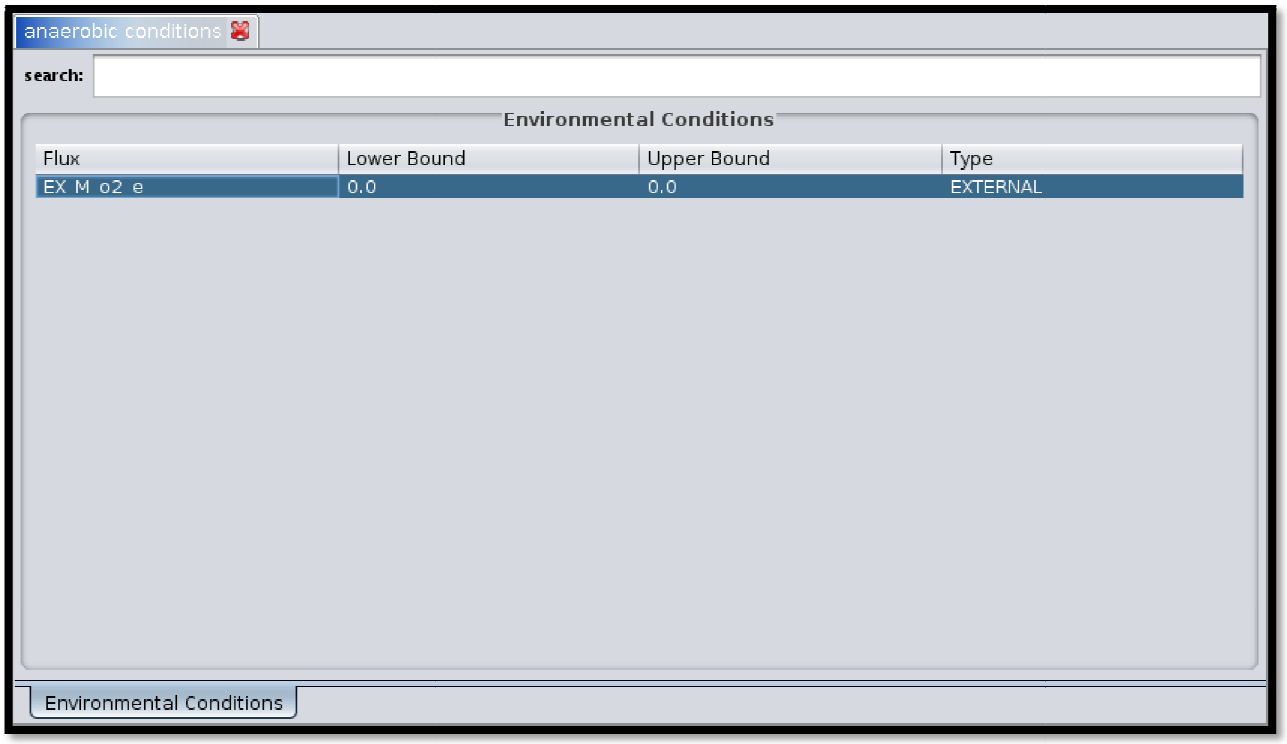
6.2.2 – EQUIVALENT VARIABLES VIEWER

Name	Equivalent Variables Viewer
Type	Viewer
Description	In this Viewer, a complete list of equivalent fluxes, as identified by *OptFlux* is presented to the user.
Components	
Search	AYT Search. Search for the desired flux in the list.
Equivalences	The list of equivalent variables.



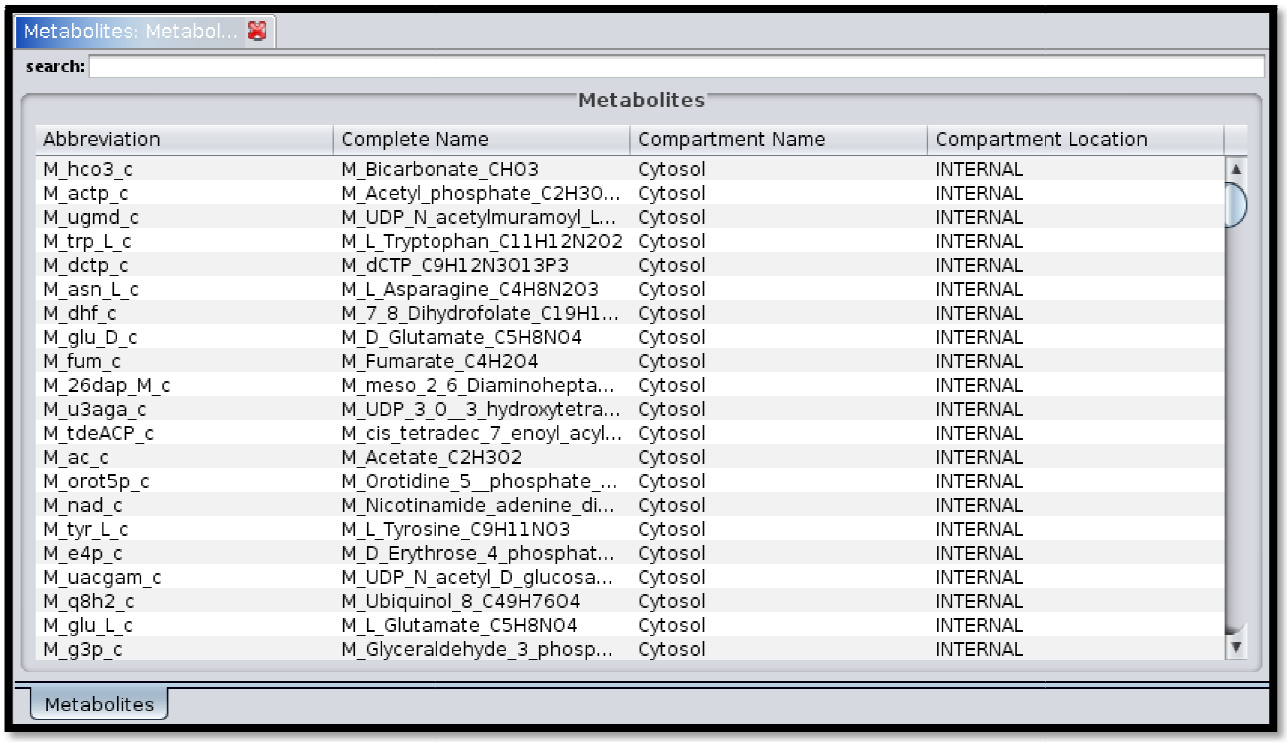
6.2.3 – Environmental Conditions Viewer

Name	Reactions Viewer
Type	Viewer
Description	The list of modified flux bounds for a given Environmental Condition is presented.
Components	
Search	AYT Search. Search for the desired flux in the list.
Fluxes	The list of the fluxes themselves.



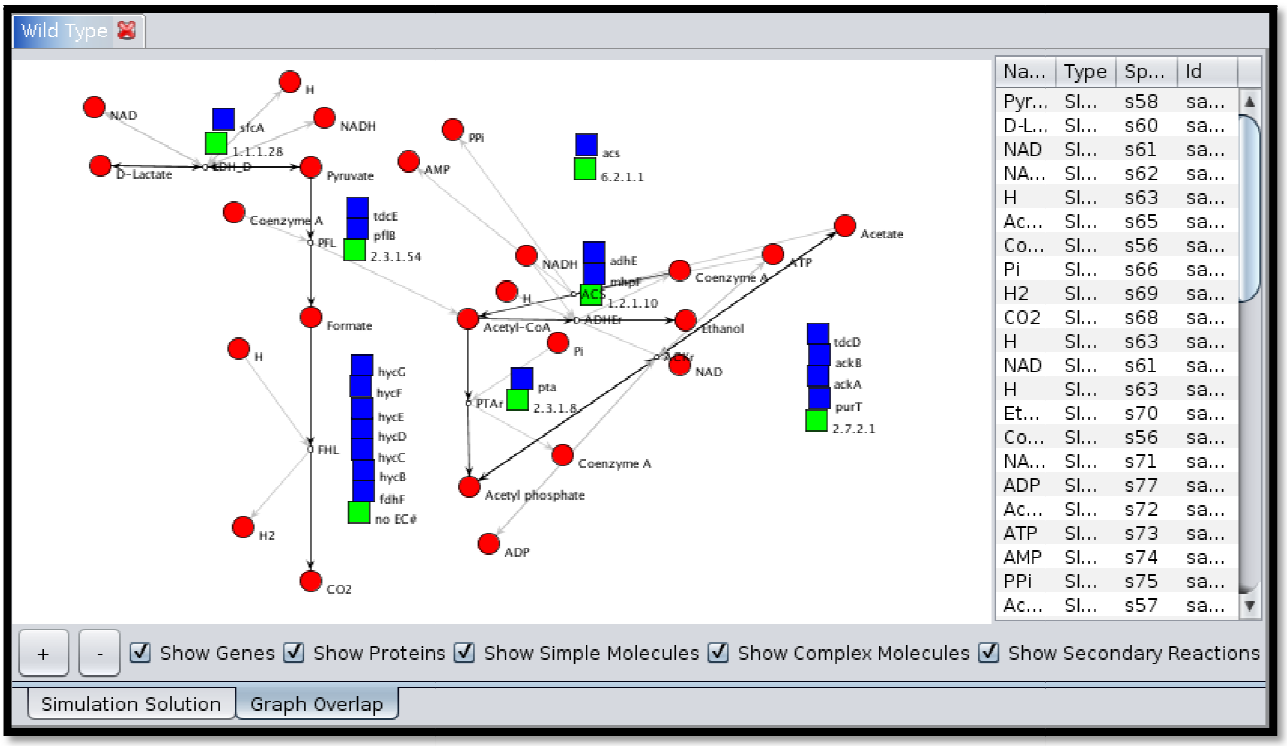
6.2.4 – METABOLITES VIEWER

Name	Metabolites Viewer
Type	Viewer
Description	This Viewer shows the metabolites abbreviations, names, compartment names and compartment locations.
Components	
Search	AYT Search. Search for the desired metabolites in the list.
Metabolites	The list of the metabolites themselves.



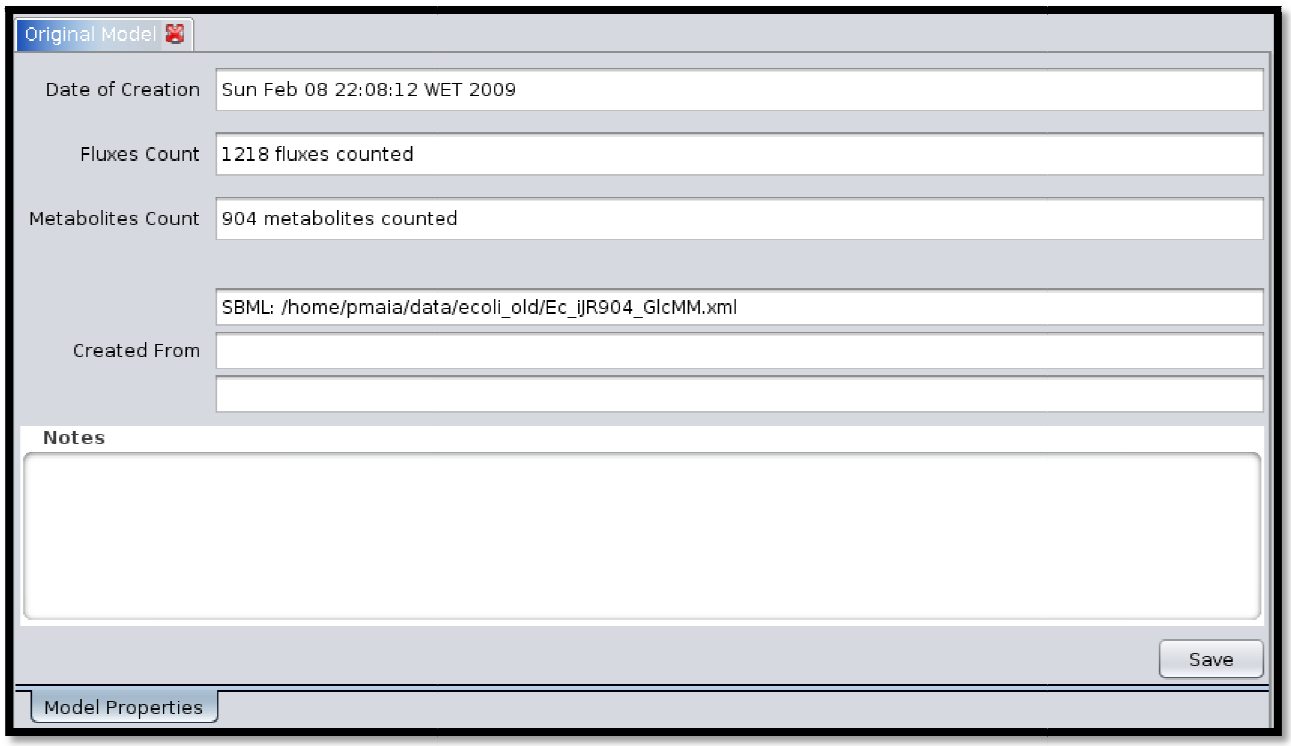
6.2.5 – Model Graph Viewer / Graph Overlap

Name	Model Graph Viewer
Type	Viewer
Description	This Viewer provides a graphical visualization of the previously loaded model in the form of a graph. The Graph Overlap Viewer is exactly the same as the Model Graph Viewer but it only appears in the visualization of a SimulationSolution. It provides an overlap of the simulation values and the Model Graph.
Components	
[+]	Zooms in (MOUSE WHEEL DOWN has the same effect).
[-]	Zooms out (MOUSE WHEEL UP has the same effect).
Show Genes	Enable/Disable the visualization of genes in the graph.
Show Proteins	Enable/Disable the visualization of proteins in the graph.
Show Simple Molecules	Enable/Disable the visualization of simple molecules in the graph.
Show Complex Molecules	Enable/Disable the visualization of complex molecules in the graph.
Show Secondary Reactions	Enable/Disable the visualization of the secondary reactions in the graph.
Table	A list of all the components shown in the pathway



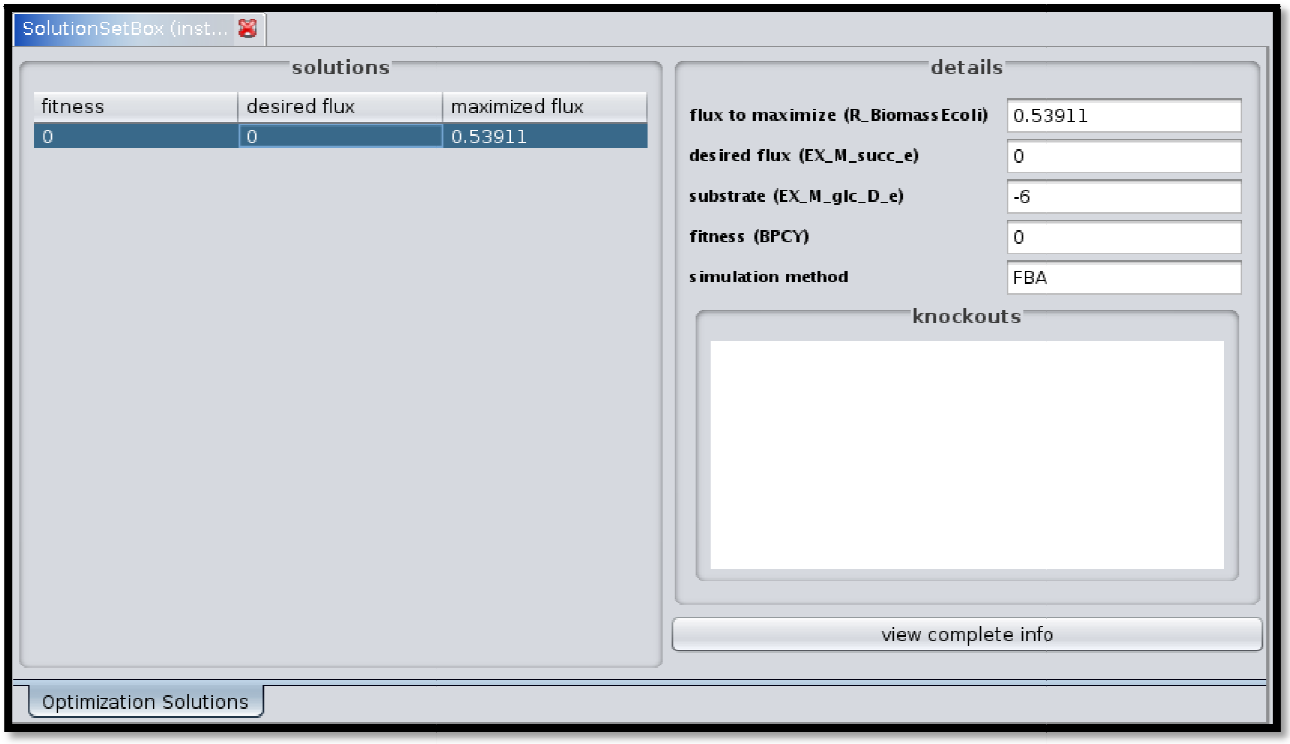
6.2.6 – Model Properties Viewer

Name	Model Properties Viewer
Type	Viewer / Editor
Description	This Viewer is also an Editor. It allows for the insertion of notes relative to the metabolic model.
Components	
Date of Creation	The date when the Model was created. Equals the creation date of the Project.
Fluxes Count	The number of fluxes found in the Model.
Metabolites Count	The number of metabolites found in the Model.
Created From	The paths of the files from which the Model was created.
Annotations	Optional annotations that the user can add to better describe the Model.
Save	Saves the modifications (if any) that the user made in the notes area.



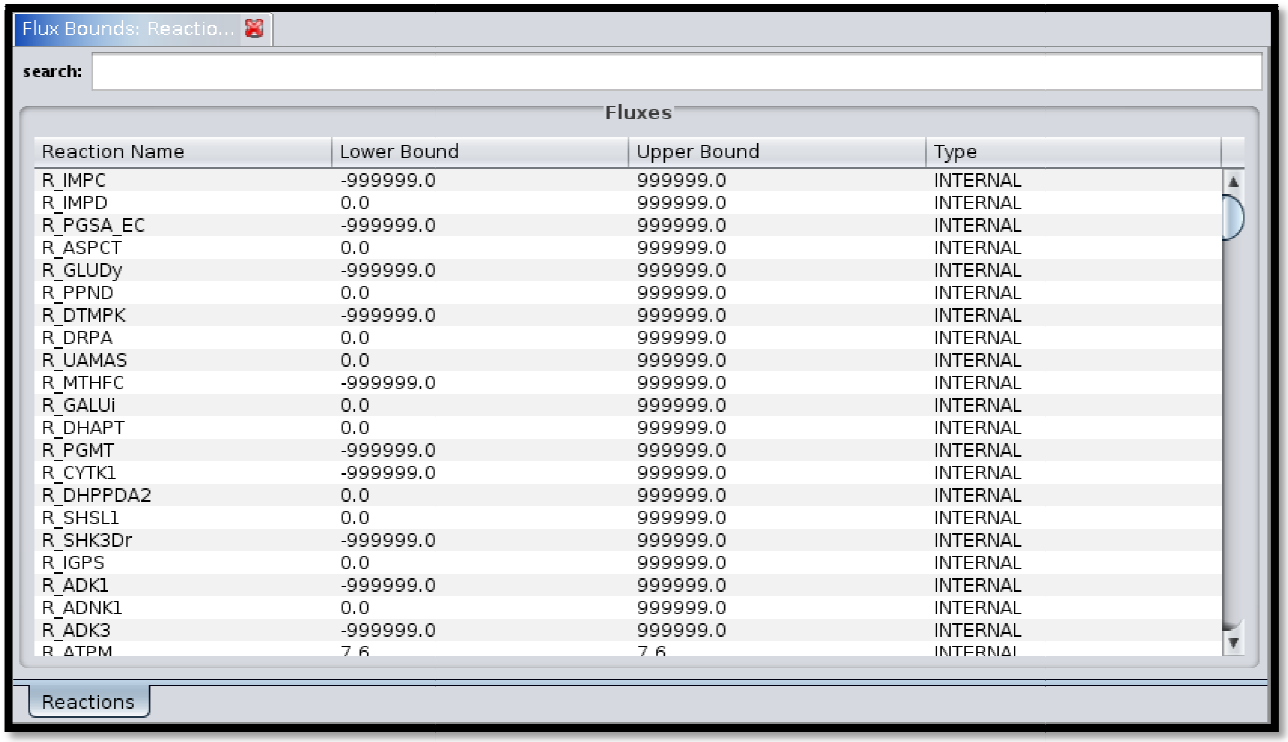
6.2.7 – Optimization Solutions Viewer

Name	Optimization Solutions Viewer
Type	Viewer
Description	This Viewer provides a list of all the different solutions gathered during an optimization procedure. Only different solutions are kept and they are sorted by the best fitness.
Components	
Solutions	The list of solutions kept after the optimization procedure.
Details	In this panel, some details about the solution selected in the [solutions] list, are shown. Specifically, the name and value of: the maximized flux, the desired flux, the substrate flux, the fitness function and the simulation method.
Knockouts	The list of knocked-out fluxes in the solution selected from the [solutions] list.
View Complete Info	Launches a Simulation Solution Viewer (see 6.2.9) containing the complete information about the selected solution.



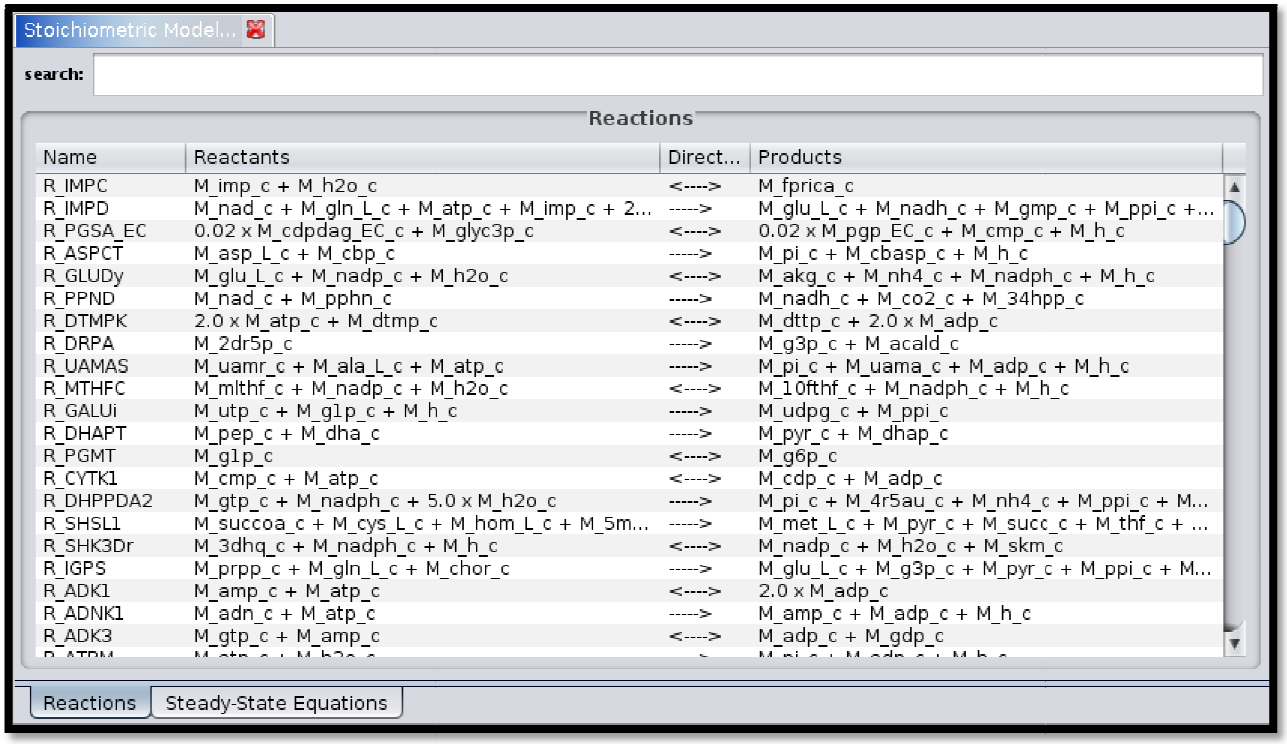
6.2.8 – Reactions Viewer (Flux Bounds)

Name	Reactions Viewer
Type	Viewer / Editor
Description	This Viewer is also an Editor. It allows the selection of the type of the flux, INTERNAL->EXTERNAL, EXTERNAL->INTERNAL.
Components	
Search	AYT Search. Search for the desired flux in the list.
Fluxes	The list of the fluxes themselves.



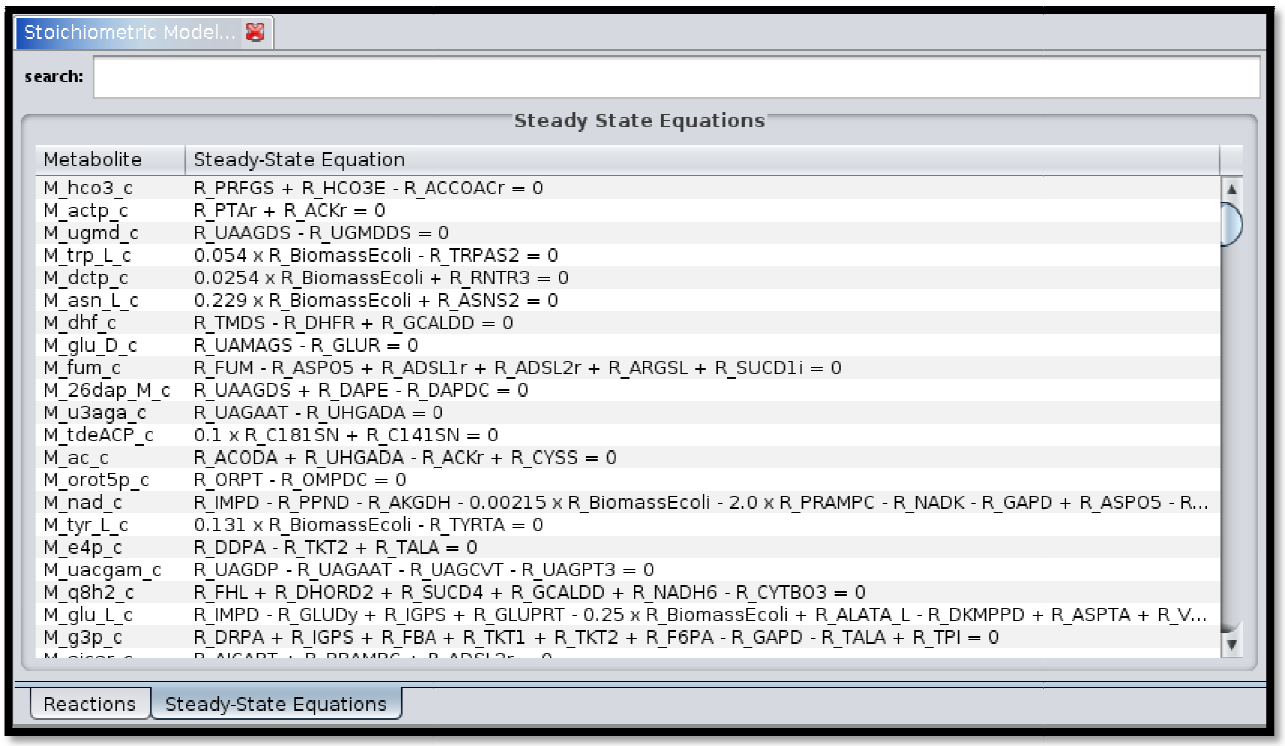
6.2.9 – Reactions Viewer (Stoichiometric Model)

Name	Reactions Viewer
Type	Viewer
Description	In here, a full view of the Stoichiometric Matrix is presented. The Metabolites and its coefficients are presented in the form of equations for each different flux. The Reactants, Products and the direction of the reaction are all properly identified.
Components	
Search	AYT Search. Search for the desired flux or metabolite in the list.
Reactions	The complete list of reactions present in the model.


Stoichiometric Model...
search:
Reactions
Name
Reactants
Direct...
Products
R_IMPC M_imp_c + M_h2o_c <----> M_fprica_c
R_IMPD M_nad_c + M_gln_L_c + M_atp_c + M_imp_c + 2... -----> M_glu_L_c + M_nadh_c + M_gmp_c + M_ppi_c +...
R_PGSA_EC 0.02 x M_cdpdag_EC_c + M_glyc3p_c <----> 0.02 x M_pgp_EC_c + M_cmp_c + M_h_c
R_ASPCT M_asp_L_c + M_cbp_c -----> M_pi_c + M_cbasp_c + M_h_c
R_GLUDy M_glu_L_c + M_nadp_c + M_h2o_c <----> M_akg_c + M_nh4_c + M_nadph_c + M_h_c
R_PPND M_nad_c + M_pphn_c -----> M_nadh_c + M_co2_c + M_34hpp_c
R_DTMPK 2.0 x M_atp_c + M_dtmp_c <----> M_dttp_c + 2.0 x M_adp_c
R_DRPA M_2dr5p_c -----> M_g3p_c + M_acald_c
R_UAMAS M_uamr_c + M_ala_L_c + M_atp_c -----> M_pi_c + M_uama_c + M_adp_c + M_h_c
R_MTHFC M_mlthf_c + M_nadp_c + M_h2o_c <----> M_10fthf_c + M_nadph_c + M_h_c
R_GALUi M_utp_c + M_g1p_c + M_h_c -----> M_udpg_c + M_ppi_c
R_DHAPT M_pep_c + M_dha_c -----> M_pyr_c + M_dhap_c
R_PGMT M_g1p_c <----> M_g6p_c
R_CYTK1 M_cmp_c + M_atp_c <----> M_cdp_c + M_adp_c
R_DHPPDA2 M_gtp_c + M_nadph_c + 5.0 x M_h2o_c -----> M_pi_c + M_4r5au_c + M_nh4_c + M_ppi_c + M...
R_SHSL1 M_succoa_c + M_cys_L_c + M_hom_L_c + M_5m... -----> M_met_L_c + M_pyr_c + M_succ_c + M_thf_c + ...
R_SHK3Dr M_3dhq_c + M_nadph_c + M_h_c <----> M_nadp_c + M_h2o_c + M_skm_c
R_IGPS M_prpp_c + M_gln_L_c + M_chor_c -----> M_glu_L_c + M_g3p_c + M_pyr_c + M_ppi_c + M...
R_ADK1 M_amp_c + M_atp_c <----> 2.0 x M_adp_c
R_ADNK1 M_adn_c + M_atp_c -----> M_amp_c + M_adp_c + M_h_c
R_ADK3 M_gtp_c + M_amp_c <----> M_adp_c + M_gdp_c
Reactions
Steady-State Equations

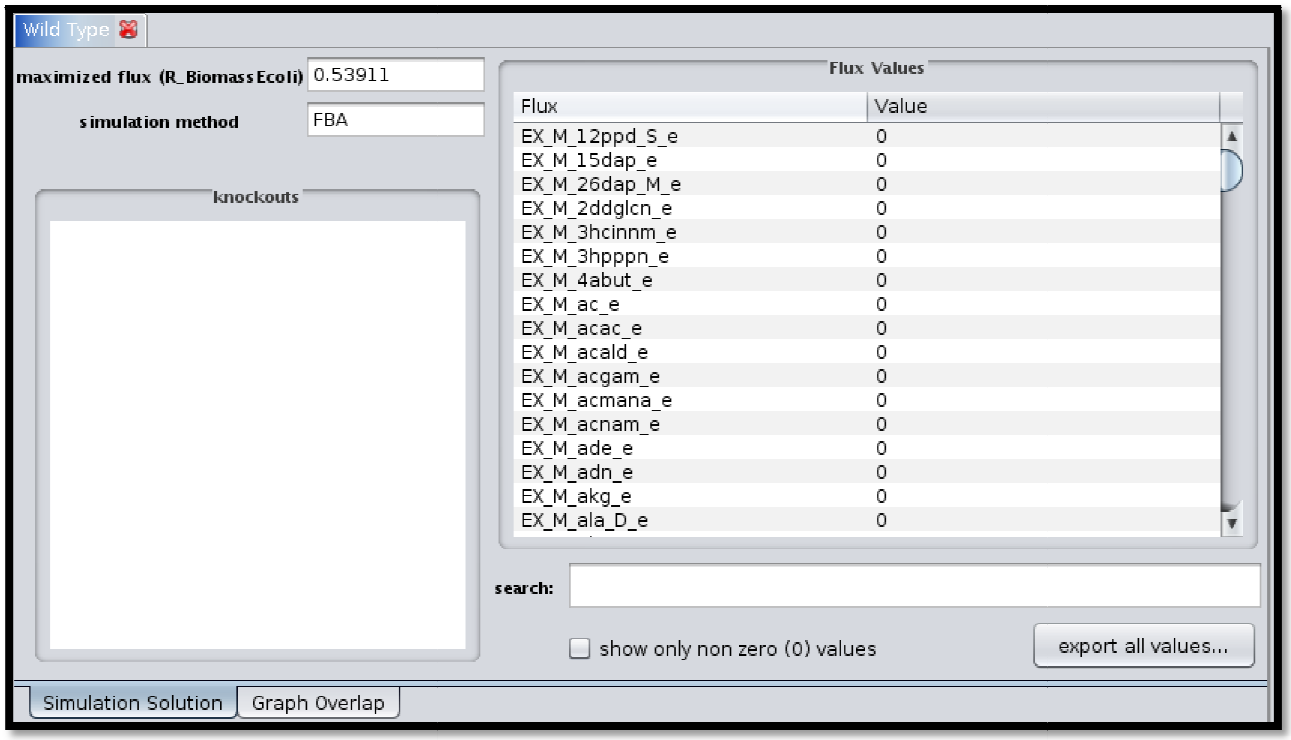
6.2.10 – Steady-State Equations Viewer (Stoichiometric Model)

Name	Steady-State Equations Viewer
Type	Viewer
Description	This view presents the stoichiometric model in the form of steady-state equations for the balanced metabolites.
Components	
Search	AYT Search. Search for the desired flux or metabolite in the list.
Steady-State Equations	The complete list of balanced metabolites and respective steady-state equations present in the model.


Stoichiometric Model...
search:
Steady State Equations
Metabolite
Steady-State Equation
M_hco3_c R_PRFGS + R_HCO3E - R_ACCOACr = 0
M_actp_c R_PTAr + R_ACKr = 0
M_ugmd_c R_UAAGDS - R_UGMDDS = 0
M_trp_L_c 0.054 x R_BiomassEcoli - R_TRPAS2 = 0
M_dctp_c 0.0254 x R_BiomassEcoli + R_RNTR3 = 0
M_asn_L_c 0.229 x R_BiomassEcoli + R_ASNS2 = 0
M_dhf_c R_TMDS - R_DHFR + R_GCALDD = 0
M_glu_D_c R_UAMAGS - R_GLUR = 0
M_fum_c R_FUM - R_ASPO5 + R_ADSL1r + R_ADSL2r + R_ARGSL + R_SUCD1i = 0
M_26dap_M_c R_UAAGDS + R_DAPE - R_DAPDC = 0
M_u3aga_c R_UAGAAT - R_UHGADA = 0
M_tdeACP_c 0.1 x R_C181SN + R_C141SN = 0
M_ac_c R_ACODA + R_UHGADA - R_ACKr + R_CYSS = 0
M_orot5p_c R_ORPT - R_OMPDC = 0
M_nad_c R_IMPD - R_PPND - R_AKGDH - 0.00215 x R_BiomassEcoli - 2.0 x R_PRAMPC - R_NADK - R_GAPD + R_ASPO5 - R...
M_tyr_L_c 0.131 x R_BiomassEcoli - R_TYRTA = 0
M_e4p_c R_DDPA - R_TKT2 + R_TALA = 0
M_uacgam_c R_UAGDP - R_UAGAAT - R_UAGCVT - R_UAGPT3 = 0
M_q8h2_c R_FHL + R_DHORD2 + R_SUCD4 + R_GCALDD + R_NADH6 - R_CYTBO3 = 0
M_glu_L_c R_IMPD - R_GLUDy + R_IGPS + R_GLUPRT - 0.25 x R_BiomassEcoli + R_ALATA_L - R_DKMPPD + R_ASPTA + R_V...
M_g3p_c R_DRPA + R_IGPS + R_FBA + R_TKT1 + R_TKT2 + R_F6PA - R_GAPD - R_TALA + R_TPI = 0
Reactions
Steady-State Equations

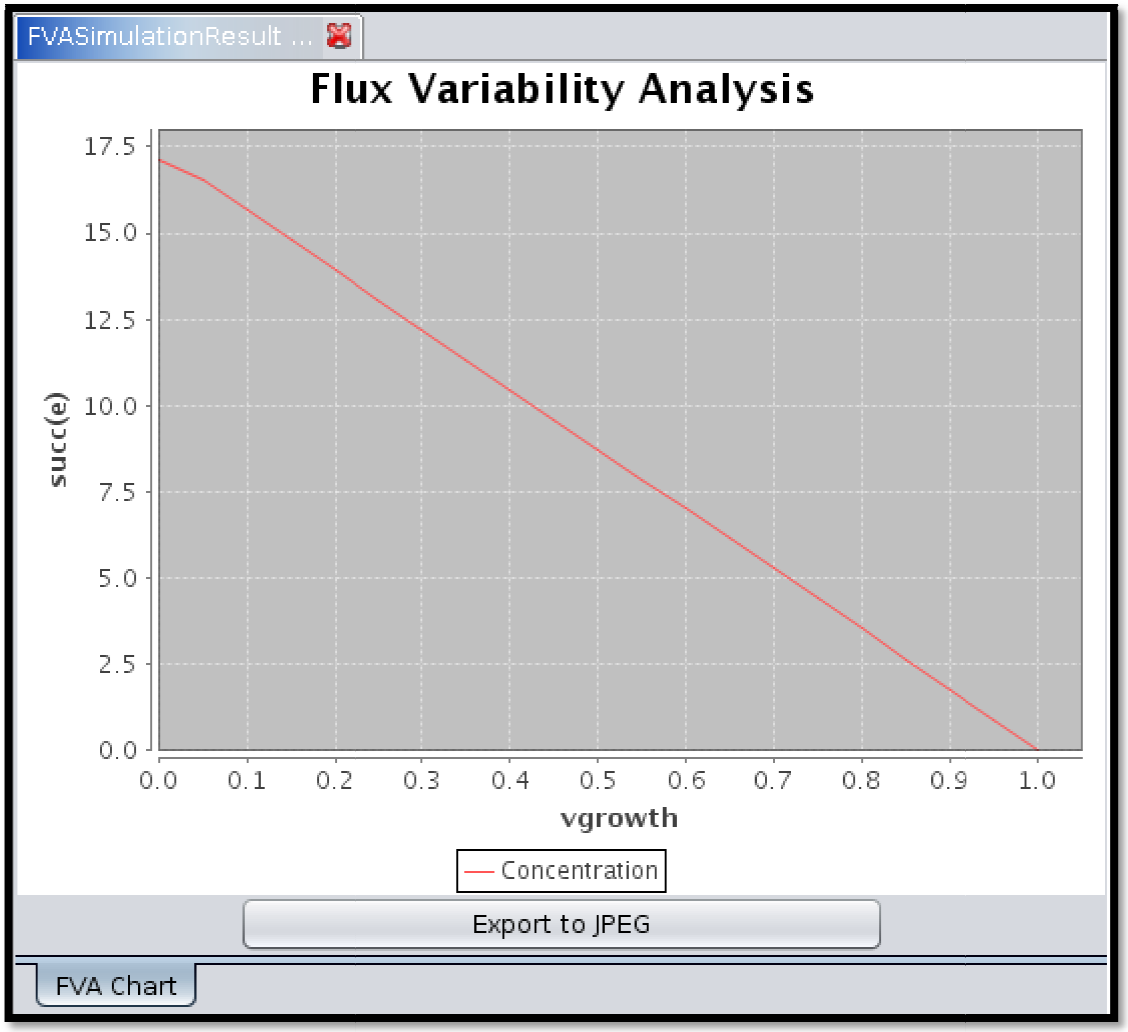
6.2.11 – SIMULATION SOLUTION VIEWER

Name	Simulation Solution Viewer
Type	Viewer
Description	This viewer shows the complete information about a certain simulation procedure. The complete list of knockouts and the values for all the fluxes in the simulated model are herein contained.
Components	
Maximized Flux	The name and value for the flux selected for maximization in this simulation.
Simulation Method	The name of the simulation method used in this simulation.
Knockouts	The list of knocked-out fluxes in this solution.
Flux Values	The values for all the fluxes of the model after the simulation procedure.
Search	AYT Search. Search for the desired flux in the list.
Export Values	Exports the above list to a text file.



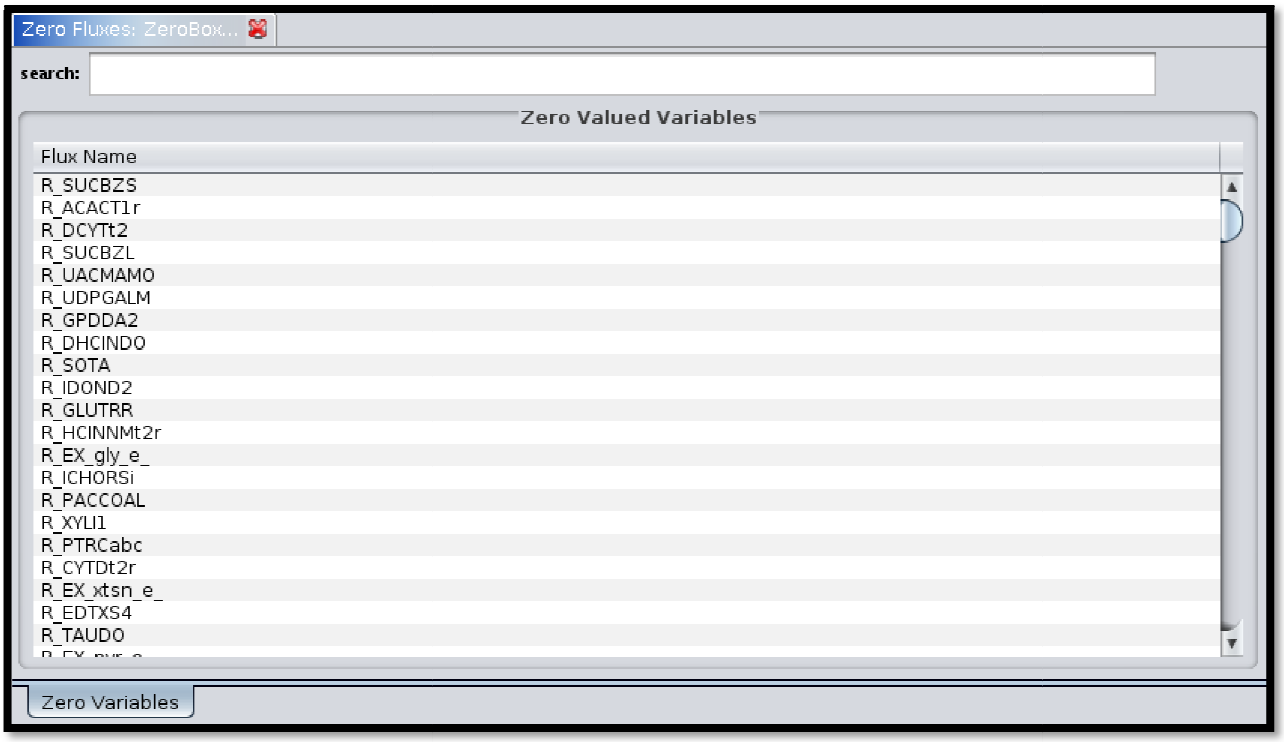
6.2.12 – VARIABILITY ANALYSIS VIEWER

Name	Variability Analysis Viewer
Type	Viewer
Description	This viewer shows a chart that represents the variability analysis process, usually of a given flux in function of the biomass growth associated flux.
Components	
Export to JPEG	Allows writing the chart to an external JPEG image.



6.2.13 – ZERO VARIABLES VIEWER

Name	Zero Variables Viewer
Type	Viewer
Description	In this Viewer, a list of fluxes that are constrained, by the linear problem, have a value equal to zero are presented.
Components	
Search	AYT Search. Search for the desired flux in the list.
Zero Valued Variables	The list of variables constrained to have value equal to zero.



7 | File Formats

This chapter explains the format of the files supported by *OptFlux*.

7.1 – Flat Files for models

7.1.1 – Flux Bounds File

For **n** reactions (R), there must be **n** lines in the file, one per each flux.
There are four fields in each line, separated by commas, tabs, or any other separator that the user decides.
First comes the **reaction name**, then the **reaction reversibility** (either **I** for Irreversible or **R** for Reversible - OPTIONAL), then the **flux lower limit** and finally the **flux upper limit**.

$R1_{Name}$	$R1_{reversibility}$	$R1_{lower\_bound}$	$R1_{upper\_bound}$
$R2_{Name}$	$R2_{reversibility}$	$R2_{lower\_bound}$	$R2_{upper\_bound}$
.	.	.	.
.	.	.	.
.	.	.	.
Rn_{Name}	$Rn_{reversibility}$	Rn_{lower_bound}	Rn_{upper_bound}

7.1.2 – Sparse Matrix File

Each line defines the presence of given metabolite (M) in a reaction (R) and the correspondent stoichiometric coefficient (Coeff). Each line is composed of three elements, separated by any separator the user wishes. First comes the **metabolite index** (in the metabolites file), then the **reaction index** (in the flux bounds file) and finally the **stoichiometric coefficient**.

$M0_{index}$	$R0_{index}$	M0_R0_Coeff
$M0_{index}$	$R10_{index}$	M0_R10_Coeff
$M0_{index}$	$R33_{index}$	M0_R33_Coeff
.	.	.
.	.	.
.	.	.
Mn_{index}	Rx_{index}	Mn_Rx_Coeff

If a metabolite appears in five reactions, there will be five lines for that metabolite, one for each reaction. Note that some users find more intuitive the indexation beginning in 1 instead of 0. OptFlux supports that option also.

7.1.3 – Full Matrix File

Each line represents a **metabolite (M)** and each column represents **a reaction (R)**. A given **position (M,R)** represents the **M-th metabolite** in the metabolites file (if one exists), and R the **R-th reaction** in the fluxes file. The value for that position is the correspondent **stoichiometric coefficient**. The fields are separated by any separator the user wishes.

M1_R1_Coeff	**M1_R2_Coeff**	**…**	**M1_Rm_Coeff**
M2_R1_Coeff	**M2_R2_Coeff**	**…**	**M2_Rm_Coeff**
M3_R1_Coeff	**M3_R2_Coeff**	**…**	**M3_Rm_Coeff**
.	**.**	**.**	**.**
.	**.**	**.**	**.**
.	**.**	**.**	**.**
Mn_R1_Coeff	**Mn_R2_Coeff**	**…**	**Mn_Rm_Coeff**

7.1.4 – Metabolites File

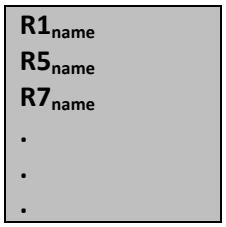
The file format for the metabolites file is very simple. For each metabolite there will be a line in this file. **Each line can have up to 4 components**. Each component is separated by tabs (\t) or any other separator. The first position contains the **metabolite abbreviation** (if the file exists, this is the only one that is necessary), the second one contains the **complete name of the metabolite**. The third and fourth positions contain the **compartment name** and **location** respectively.

$M1_{abbreviation}$	**$M1_{name}$**	**$M1_{compName}$**	**$M1_{compLocation}$**
$M2_{abbreviation}$	**$M2_{name}$**	**$M2_{compName}$**	**$M2_{compLocation}$**
$M3_{abbreviation}$	**$M3_{name}$**	**$M3_{compName}$**	**$M3_{compLocation}$**
.	**.**	**.**	**.**
.	**.**	**.**	**.**
.	**.**	**.**	**.**
$Mn_{abbreviation}$	**Mn_{name}**	**$Mn_{compName}$**	**$Mn_{compLocation}$**

7.2 – Flat Files for Other Components

7.2.1 – Critical Reactions File

OptFlux allows users to load the critical reactions information from a previously saved or manually created file. The format of this file is the following: each line in the file must have the name one reaction that is critical, that is, for **m reactions** there will be **m lines** in the file. When manually editing these files, special attention must be taken, since all the names in this file must be present in the flux bound files, otherwise, *OptFlux* will ignore them.



7.3 – SBML files

7.3.1 – Pure SBML Files

OptFlux uses **libSBML** and, therefore, can support any SBML file that **libSBML** supports. In the current version, **SBML level 2 version 3** is the latest supported, though it handles any prior version.
To access the SBML specifications for SBML L2V3 please refer to:

http://www.sbml.org/specifications/sbml-level-2/version-3/release-2/sbml-level-2-version-3-rel-2.pdf

7.3.2 – CellDesigner SBML Files

CellDesigner SBML specifications are not available to the public. The users must contact the authors in order to access them. www.celldesigner.org